Joris Gillis, Moritz Diehl

Hierarchical seeding for efficient sparsity pattern recovery in automatic differentiation

CSC2014, Abstract CT6

Table of Contents

Table of Contents

A crash-course in automatic differentiation (AD)

Given

$$f(x) : \mathbb{R}^m \to \mathbb{R}^n$$

- Vector-valued function
- Algorithm known
- Cost of an evaluation (time): T

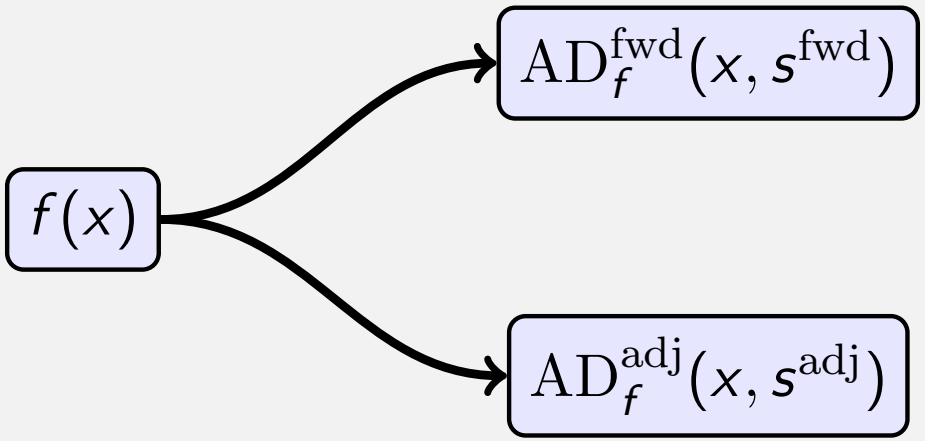
Goal

Find $J(x) = \frac{\partial f}{\partial x}(x) \in \mathbb{R}^{m \times n}$, cheaply

A crash-course in automatic differentiation (AD)

Automatic differentiation (AD)

- From a given algorithm, create two new algorithms
- Automatic procedure
- Source Code Transformation (SCT)/Operator Overloading (OO)



A crash-course in automatic differentiation (AD)

Forward sweep

$$\mathrm{AD}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}}) = J(x)s^{\mathrm{fwd}}, \qquad s^{\mathrm{fwd}} \in \mathbb{R}^n$$

Adjoint/reverse sweep

$$\mathrm{AD}_f^{\mathrm{adj}}(x, s^{\mathrm{adj}}) = J^T(x)s^{\mathrm{adj}}, \qquad s^{\mathrm{adj}} \in \mathbb{R}^m$$

Cost of one sensitivity sweep: small multiple of T

Example

n=4, m=3

$$J = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Example (forward)

n=4, m=3

$$J = \begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \end{bmatrix}$$

Forward sweep

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Example (forward)

$\mathrm{AD}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & ? & ? & ? \\ 2 & ? & ? & ? \\ 0 & ? & ? & ? \end{bmatrix}$$

Example (forward)

$\mathrm{AD}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & ? & ? \\ 2 & 0 & ? & ? \\ 0 & 5 & ? & ? \end{bmatrix}$$

Example (forward)

$\mathrm{AD}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 7 \\ 1 \\ 9 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & 7 & ? \\ 2 & 0 & 1 & ? \\ 0 & 5 & 9 & ? \end{bmatrix}$$

Example (forward)

$\mathrm{AD}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & 7 & 4 \\ 2 & 0 & 1 & 0 \\ 0 & 5 & 9 & 0 \end{bmatrix}$$

Example (reverse)

n=4, m=3

$$J = \begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \end{bmatrix}$$

Reverse sweep

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ d \\ g \\ j \end{bmatrix}$$

Composing Jacobians (reverse)

$\mathrm{AD}_f^{\mathrm{adj}}(x, s^{\mathrm{adj}})$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{Reverse sweep}} \begin{bmatrix} 0 \\ 0 \\ 7 \\ 4 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & 7 & 4 \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Composing Jacobians (reverse)

$\mathrm{AD}_f^{\mathrm{adj}}(x, s^{\mathrm{adj}})$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \xrightarrow{\text{Reverse sweep}} \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & 7 & 4 \\ 2 & 0 & 1 & 0 \\ ? & ? & ? & ? \end{bmatrix}$$

Composing Jacobians (reverse)

$\mathrm{AD}_f^{\mathrm{adj}}(x, s^{\mathrm{adj}})$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \xrightarrow{\text{Reverse sweep}} \begin{bmatrix} 0 \\ 5 \\ 9 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & 7 & 4 \\ 2 & 0 & 1 & 0 \\ 0 & 5 & 9 & 0 \end{bmatrix}$$

Composing Jacobians

Two basic sweeping strategies

- Forward sweeps with $\left\{ \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \right\}$: cost nT
- Adjoint sweeps with $\left\{ \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix} \right\}$: cost mT

Full Jacobian for cost $min(n, m)T$

Jacobian coloring

We can do better

$$J = \begin{bmatrix} 0 & 0 & 4 & 7 \\ 2 & 0 & 1 & 0 \\ 0 & 5 & 9 & 0 \end{bmatrix}$$

- What if these zeros are *always* zero?

Jacobian coloring

We can do better

$$J = \begin{bmatrix} & & 4 & 7 \\ 2 & & 1 & \\ & 5 & 9 & \end{bmatrix}$$

- What if these zeros are *structurally* zero?

Jacobian coloring

We can do better

$$J = \begin{bmatrix} & & 4 & 7 \\ 2 & & 1 & \\ & 5 & 9 & \end{bmatrix}$$

- What if these zeros are *structurally* zero?
- Forward sweeps with $\left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right\}$
- Since $\begin{bmatrix} & & g & j \\ b & & h & \\ & f & i & \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} b \\ f \\ 0 \\ j \end{bmatrix}$

Jacobian coloring

Key to cheaper Jacobian

Coloring of the column intersection graph

Our tool

$\mathrm{col}(J)$: unidirectional coloring of J's sparsity

Example

- $\mathrm{col}(J) = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right\}$
- $\mathrm{col}(J^T) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$

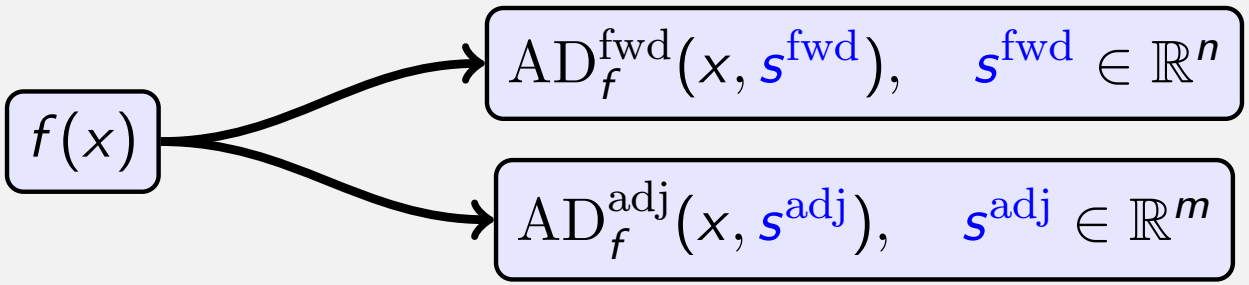
Jacobian coloring

Key question

How to find the sparsity of J?

Table of Contents

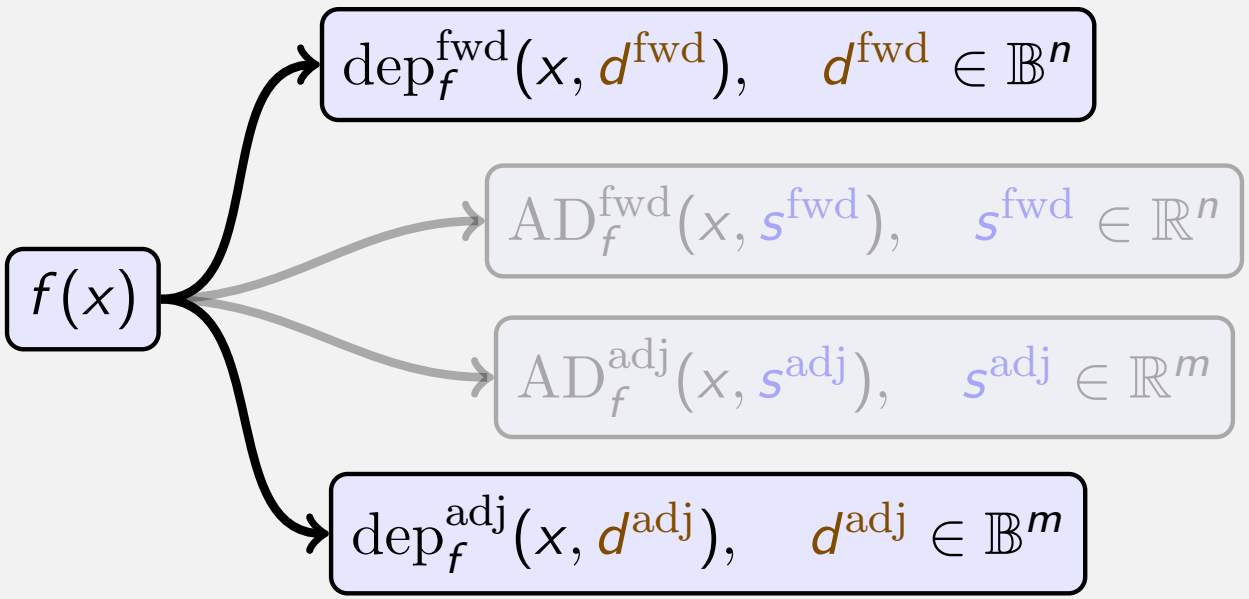
Dependency function



Dependency function

New tool in the AD toolset

- Bitvector-valued dependency function
- Cost $\tau \ll T$



Example (forward)

$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} & ? & ? & ? \\ * & ? & ? & ? \\ & ? & ? & ? \end{bmatrix}$$

Example (forward)

$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J = \begin{bmatrix} & & ? & ? \\ * & & ? & ? \\ & * & ? & ? \end{bmatrix}$$

Example (forward)

$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$J = \begin{bmatrix} & & * & ? \\ * & & * & ? \\ & * & * & ? \end{bmatrix}$$

Example (forward)

$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} & & * & * \\ * & & * & \\ & * & * & \end{bmatrix}$$

Composing Jacobian sparsity

Two basic sweeping strategies

- Forward sweeps with $\left\{ \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \right\}$: cost $n\tau$
- Adjoint sweeps with $\left\{ \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix} \right\}$: cost $m\tau$

Full Jacobian *sparsity* for cost $min(n, m)\tau$

A familiar pattern

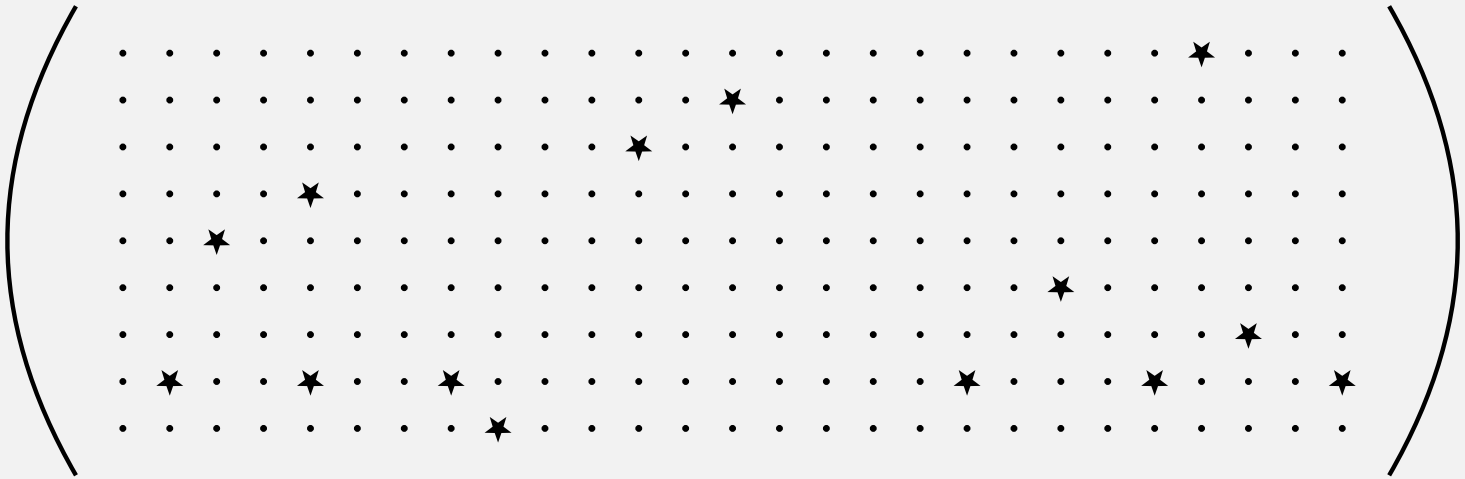
Can we do better?

Table of Contents

Obtaining sparsity (hierarchical)

New example

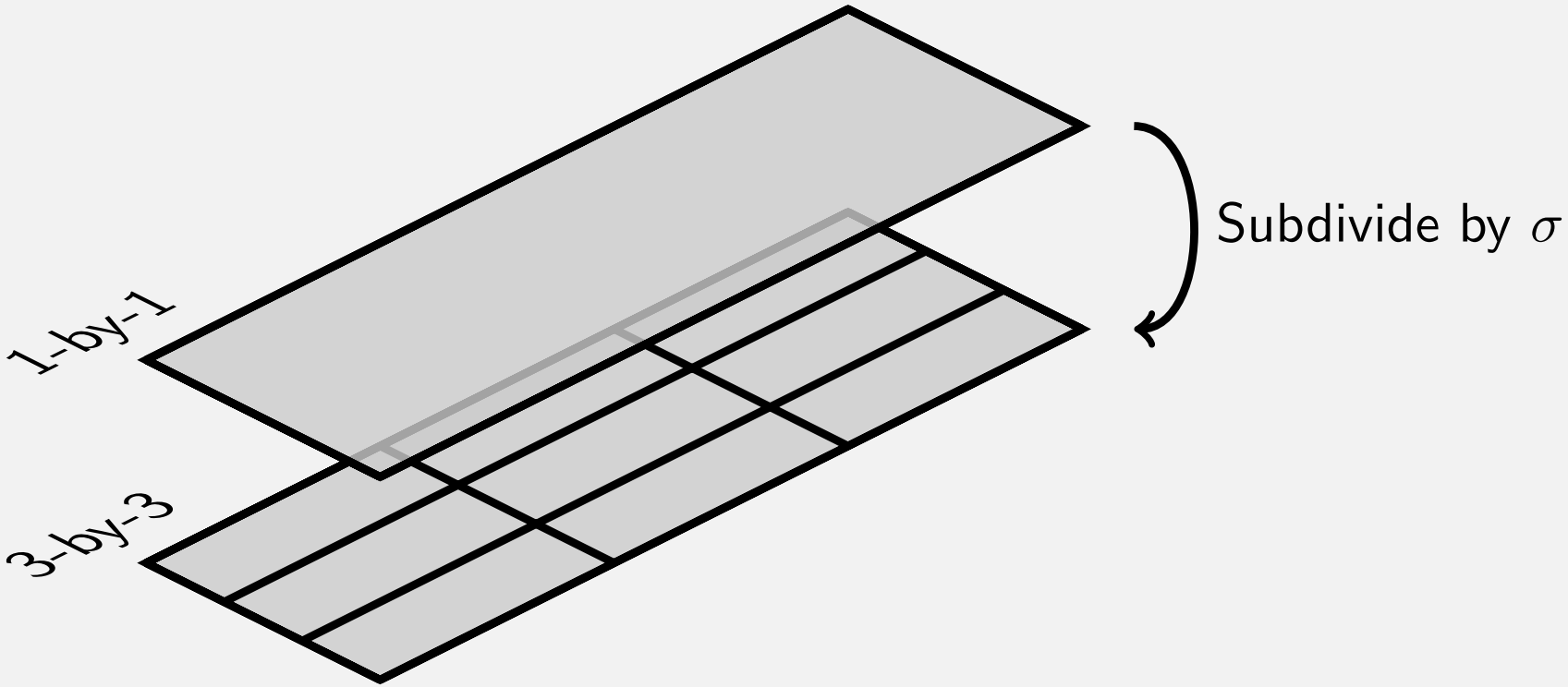
m=9 , n=27



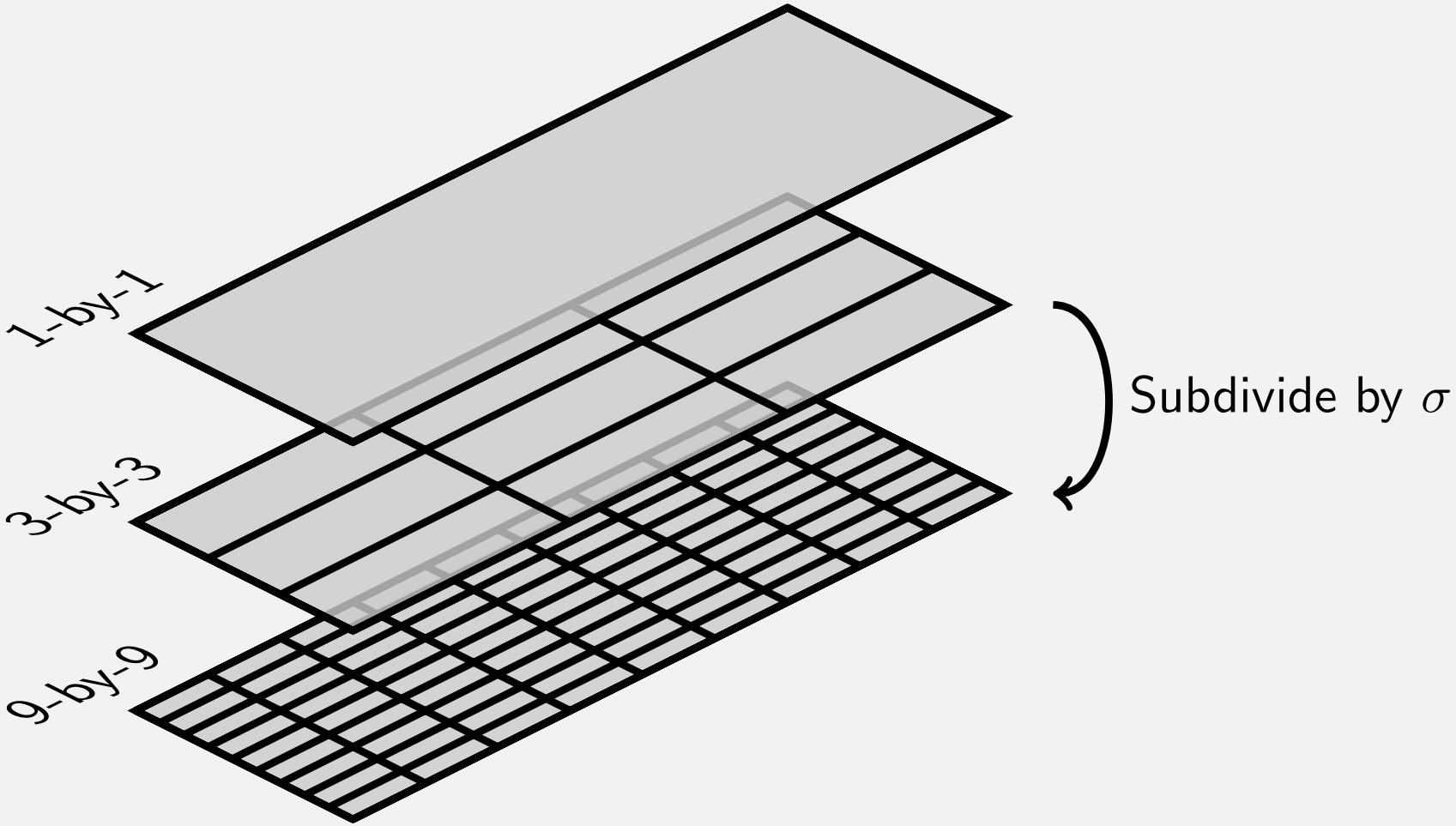
Mission

Obtain the above sparsity with less than 9 dependency sweeps.

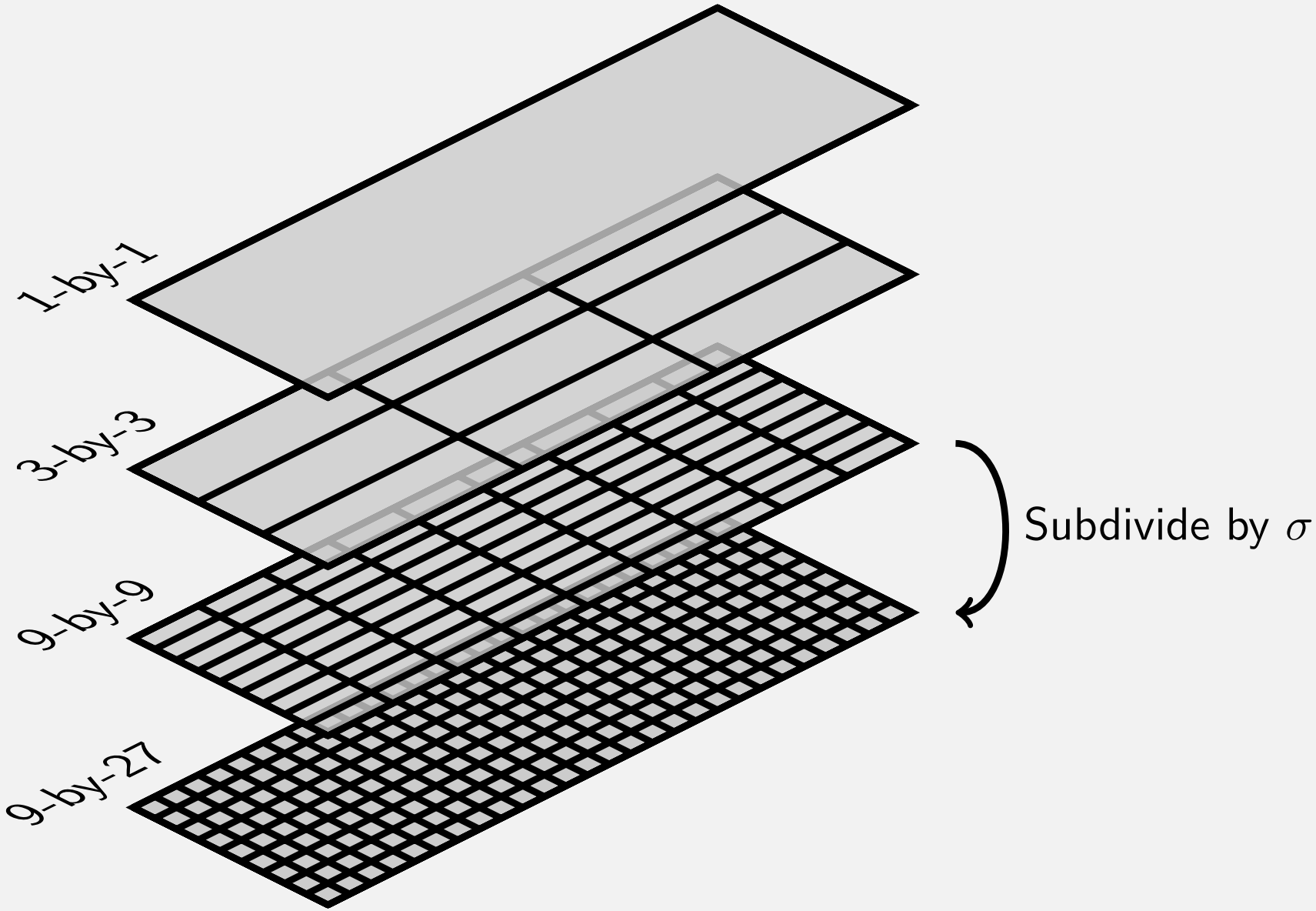
Obtaining sparsity (hierarchical)



Obtaining sparsity (hierarchical)

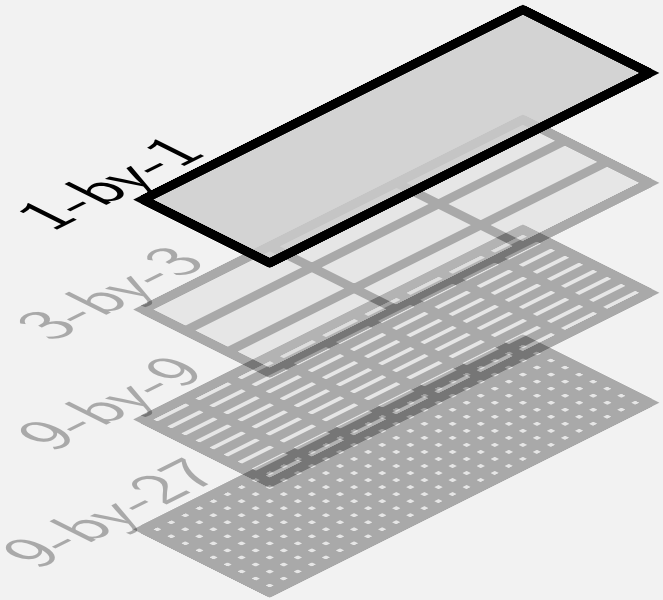


Obtaining sparsity (hierarchical)



Obtaining sparsity (hierarchical)

Step 0 (trivial)



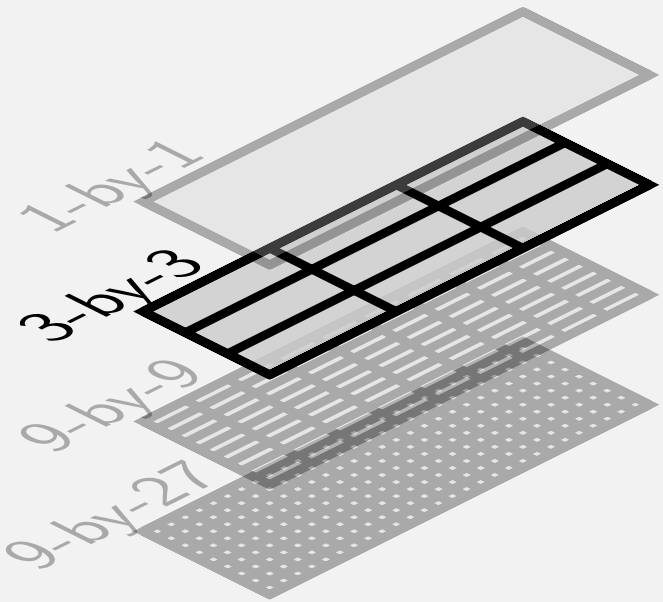
$$r = \begin{pmatrix} \star \end{pmatrix}$$

Interpretation

Assume worst-case: there is at least one non-zero in J

Obtaining sparsity (hierarchical)

Step 1



$$r = \begin{pmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{pmatrix}$$

Obtaining sparsity (hierarchical)

Step 1

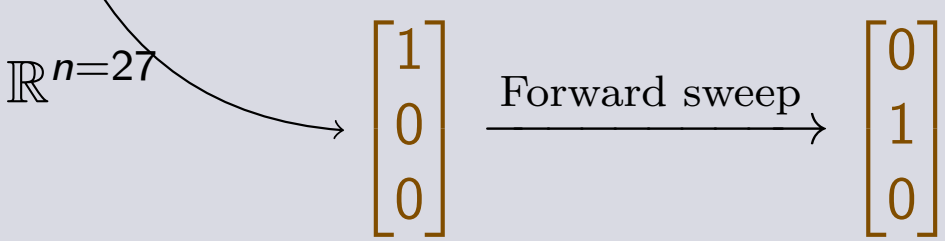
$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{Forward sweep}} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Obtaining sparsity (hierarchical)

Step 1

$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$



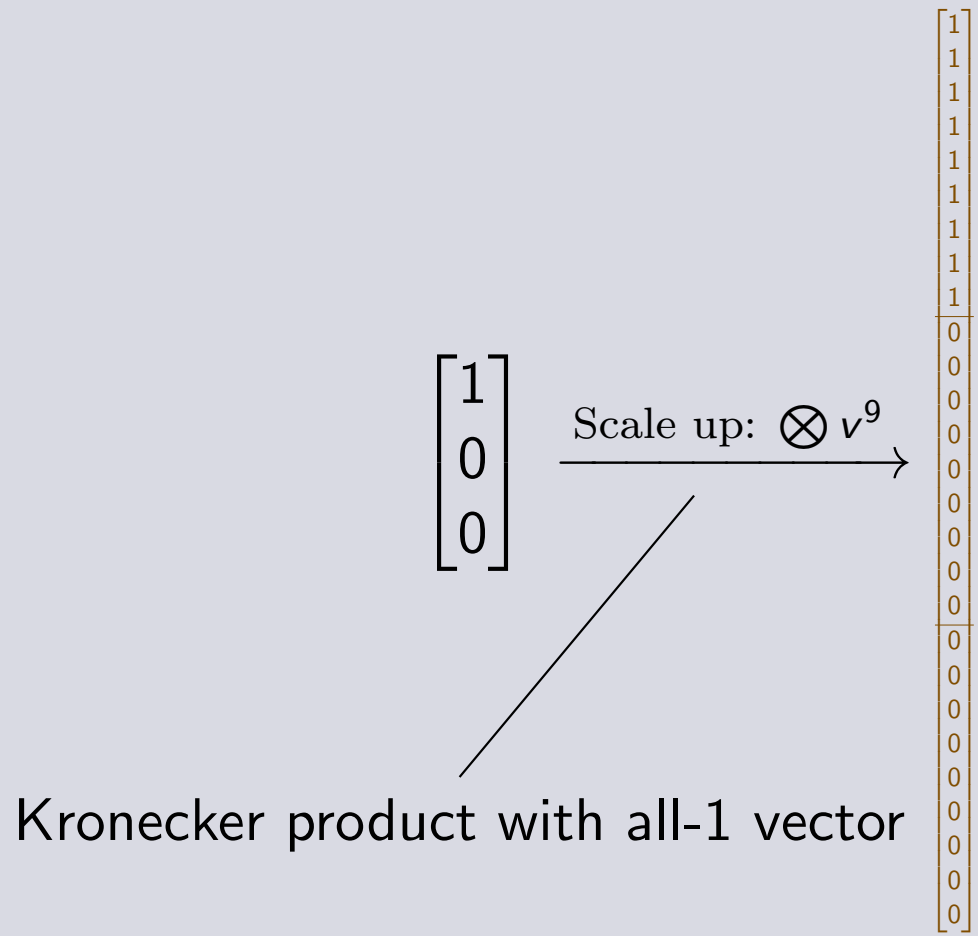
Doesn't work

Dimensions mismatch

Obtaining sparsity (hierarchical)

Step 1

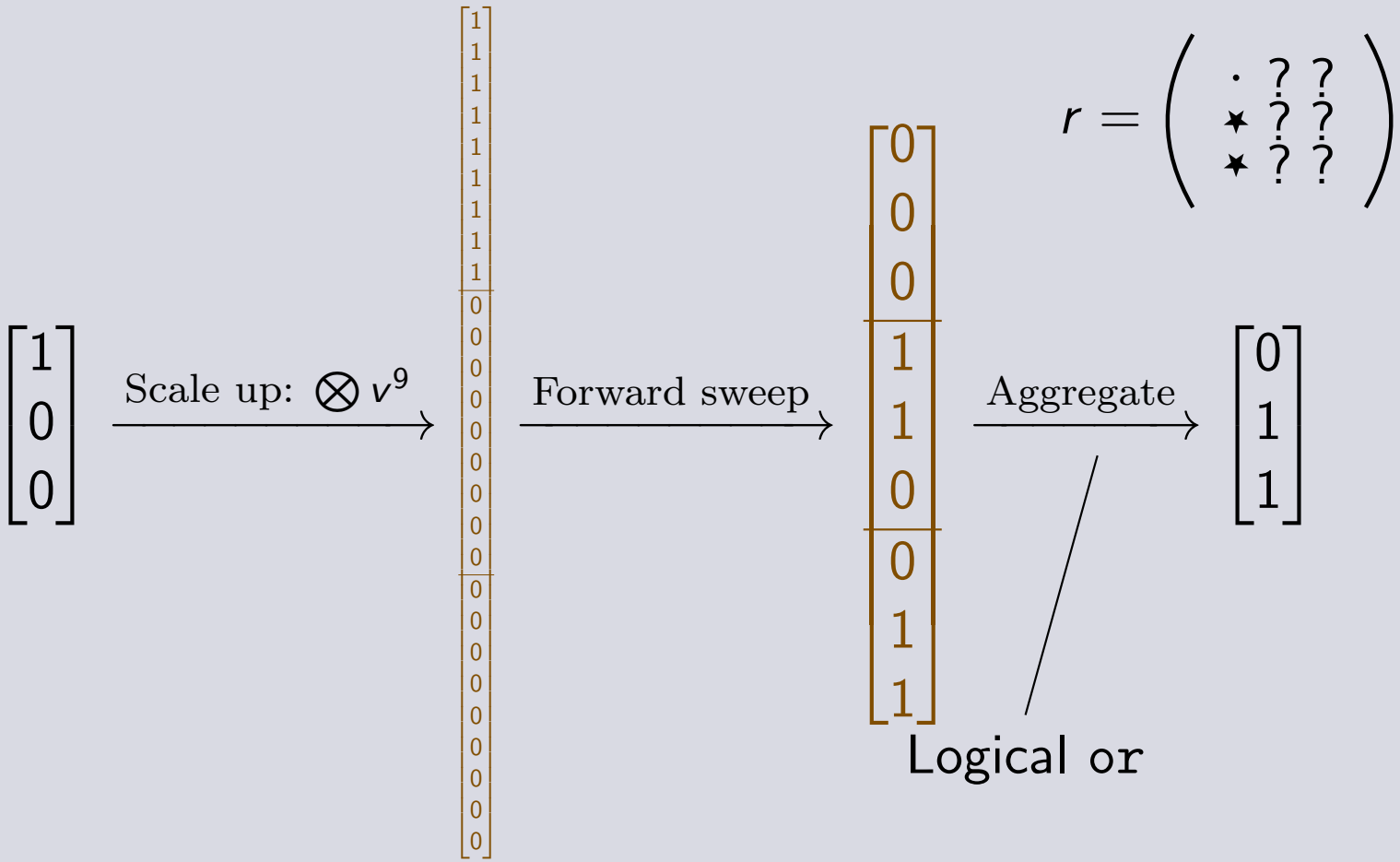
$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$



Obtaining sparsity (hierarchical)

Step 1

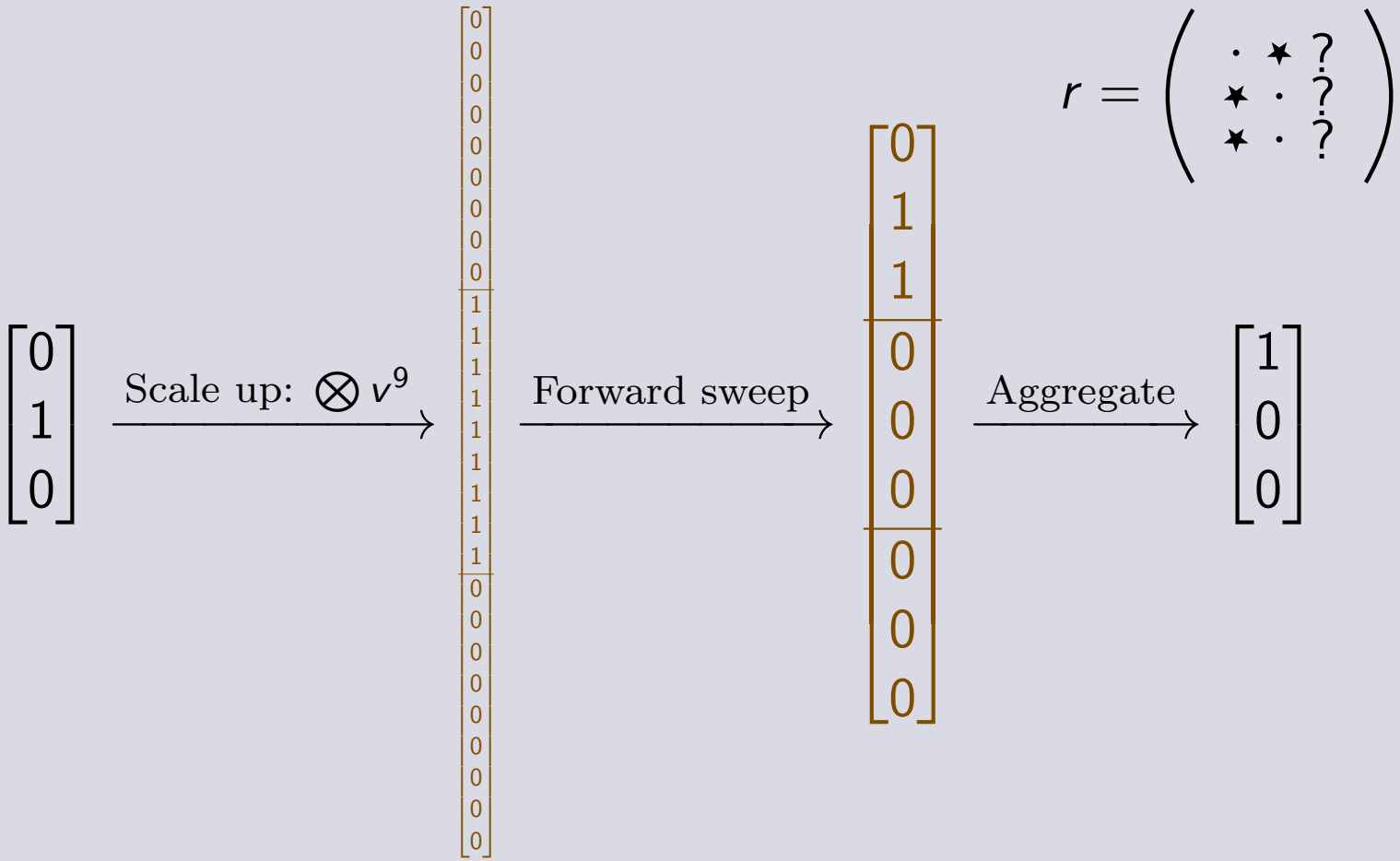
$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$



Obtaining sparsity (hierarchical)

Step 1

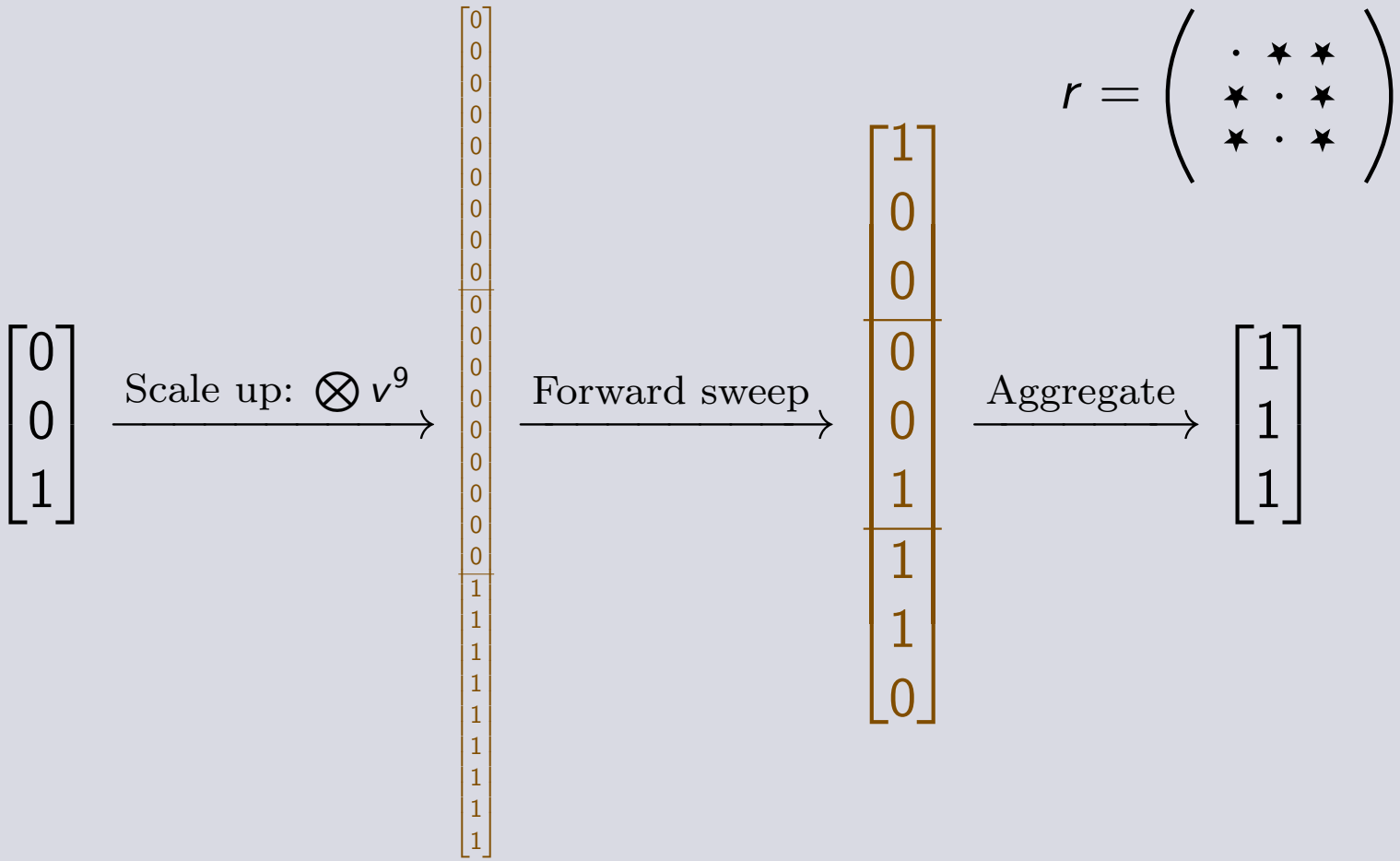
$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$



Obtaining sparsity (hierarchical)

Step 1

$\mathrm{dep}_f^{\mathrm{fwd}}(x, s^{\mathrm{fwd}})$



Step 1

More compact notation

- Block sparsity seeding: dependency sweeps with columns of

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right]$$

Step 1

More compact notation

- Block sparsity seeding: dependency sweeps with columns of

$$\mathbf{1}^3 \otimes v^9 = \begin{bmatrix} 1&0&0\\1&0&0\\1&0&0\\1&0&0\\1&0&0\\1&0&0\\1&0&0\\1&0&0\\1&0&0 \\ \hline 0&1&0\\0&1&0\\0&1&0\\0&1&0\\0&1&0\\0&1&0\\0&1&0\\0&1&0\\0&1&0 \\ \hline 0&0&1\\0&0&1\\0&0&1\\0&0&1\\0&0&1\\0&0&1\\0&0&1\\0&0&1\\0&0&1 \end{bmatrix}$$ result in bitmatrix $d =$ $$\left[\begin{array}{ccc} 0&0&1\\0&1&0\\0&1&0 \\ \hline 1&0&0\\1&0&0\\0&0&1 \\ \hline 0&0&1\\1&0&1\\1&0&0 \end{array}\right]$$

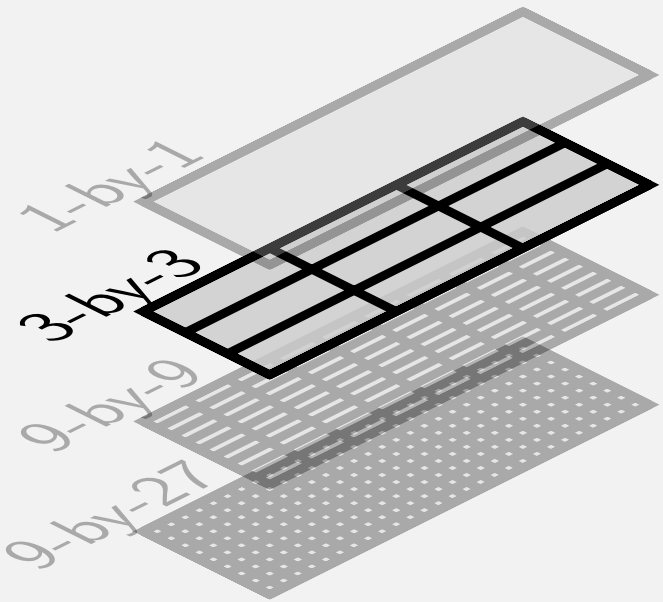
Step 1

More compact notation

- Block sparsity seeding: dependency sweeps with columns of $\mathbf{1}^3 \bigotimes v^9$ result in bitmatrix d
- Block sparsity aggregate: $(\mathbf{1}^3 \bigotimes h^3) \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

Obtaining sparsity (hierarchical)

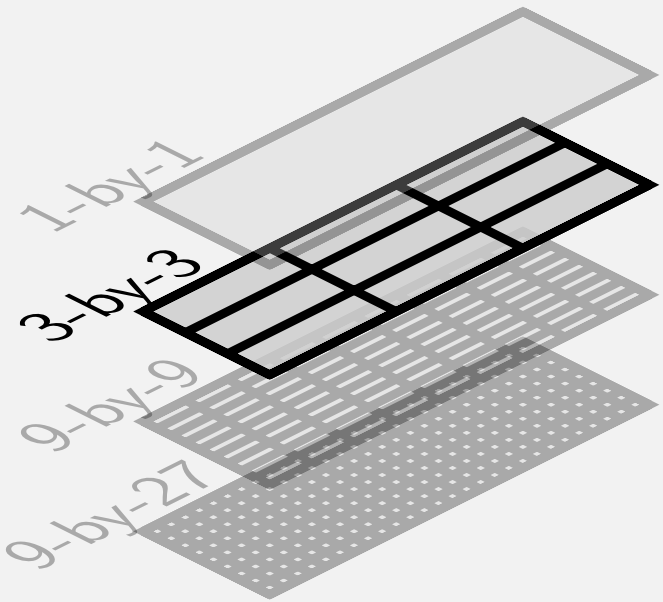
Step 1



$$r = \begin{pmatrix} \cdot & \star & \star \\ \star & \cdot & \star \\ \star & \cdot & \star \end{pmatrix}$$

Obtaining sparsity (hierarchical)

Step 1



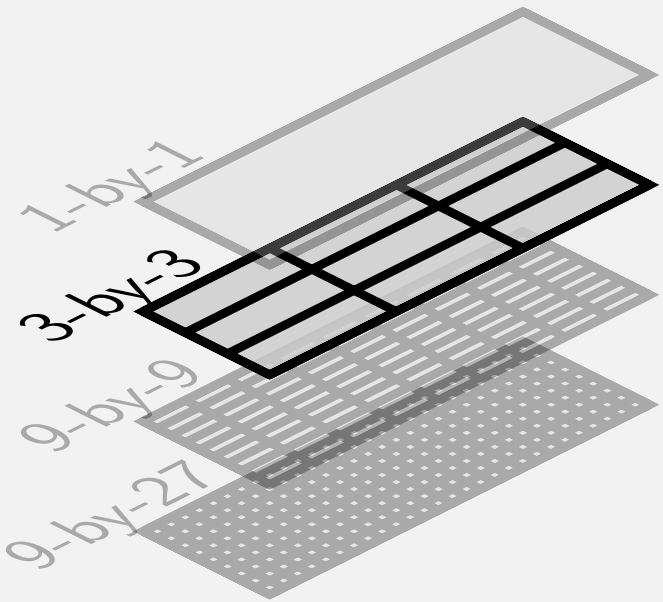
$$r = \begin{pmatrix} \cdot & \star & \star \\ \star & \cdot & \star \\ \star & \cdot & \star \end{pmatrix}$$

$$\mathrm{col}(r) = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

$$\mathrm{col}(r^T) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Obtaining sparsity (hierarchical)

Step 1



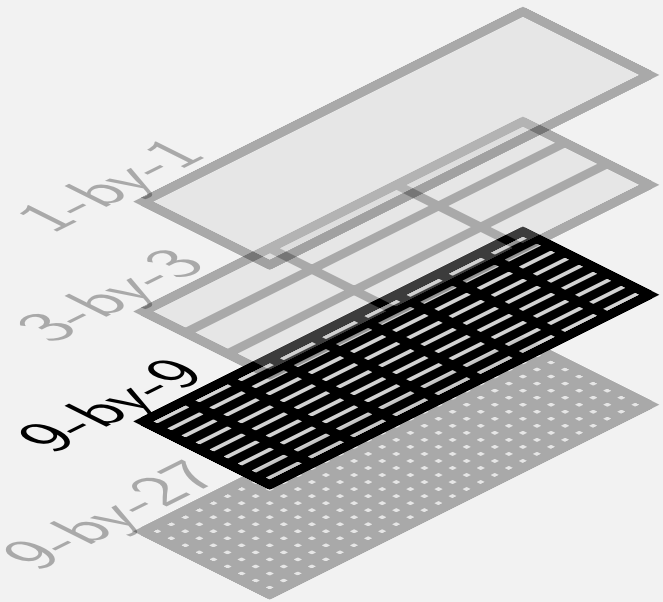
$$r = \begin{pmatrix} \cdot & \star & \star \\ \star & \cdot & \star \\ \star & \cdot & \star \end{pmatrix}$$

$$\text{seed} = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

$$\text{mode} = \text{forward}$$

Obtaining sparsity (hierarchical)

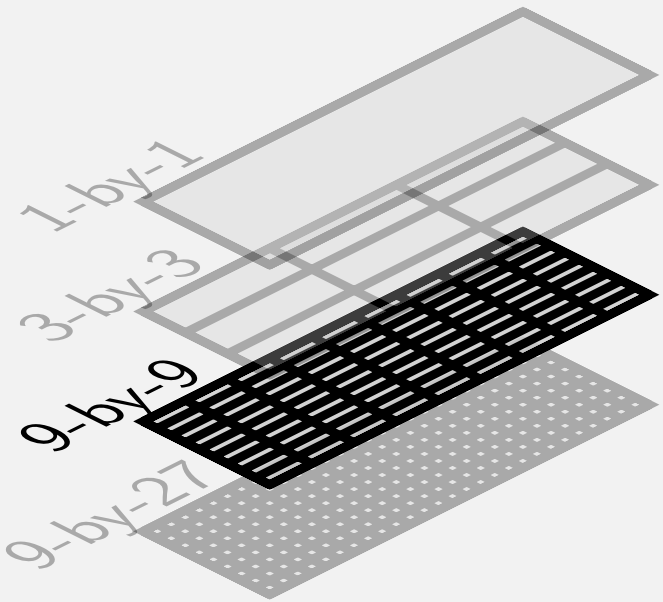
Step 2



$$r = \begin{pmatrix} ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? \end{pmatrix}$$

Obtaining sparsity (hierarchical)

Step 2



$$r = \begin{pmatrix} \cdot & \cdot & \cdot & ? & ? & ? & ? & ? & ? \\ \cdot & \cdot & \cdot & ? & ? & ? & ? & ? & ? \\ \cdot & \cdot & \cdot & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & \cdot & \cdot & \cdot & ? & ? & ? \\ ? & ? & ? & \cdot & \cdot & \cdot & ? & ? & ? \\ ? & ? & ? & \cdot & \cdot & \cdot & ? & ? & ? \\ ? & ? & ? & \cdot & \cdot & \cdot & ? & ? & ? \\ ? & ? & ? & \cdot & \cdot & \cdot & ? & ? & ? \\ ? & ? & ? & \cdot & \cdot & \cdot & ? & ? & ? \end{pmatrix}$$

Obtaining sparsity (hierarchical)

Step 2

For each s in seed

Block sparsity seeding with $s \otimes \mathbf{1}^3 \otimes v^3$

Obtaining sparsity (hierarchical)

Step 2

$s = \text{seed}_0$

Block sparsity seeding with $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]$

Obtaining sparsity (hierarchical)

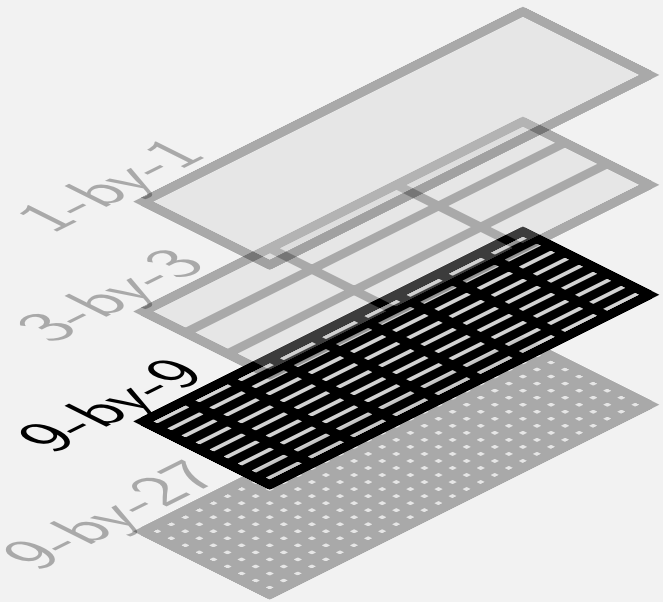
Step 2

$s = \text{seed}_0$

Block sparsity seeding with $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ results in $d = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

Obtaining sparsity (hierarchical)

Step 2



$$r = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? \\ \cdot & \cdot & \cdot & \cdot & \star & \cdot & ? & ? & ? \\ \cdot & \cdot & \cdot & \star & \cdot & \cdot & ? & ? & ? \\ \cdot & \star & \cdot & \cdot & \cdot & \cdot & ? & ? & ? \\ \star & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? \\ \star & \star & \star & \cdot & \cdot & \cdot & ? & ? & ? \\ \cdot & \cdot & \star & \cdot & \cdot & \cdot & ? & ? & ? \end{pmatrix}$$

$$d = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{array}\right]$$

Obtaining sparsity (hierarchical)

Step 2

$s = \text{seed}_1$

Block sparsity seeding with $s \otimes \mathbf{1}^3 \otimes v^3$

Obtaining sparsity (hierarchical)

Step 2

$s = \text{seed}_1$

Block sparsity seeding with $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right]$

Obtaining sparsity (hierarchical)

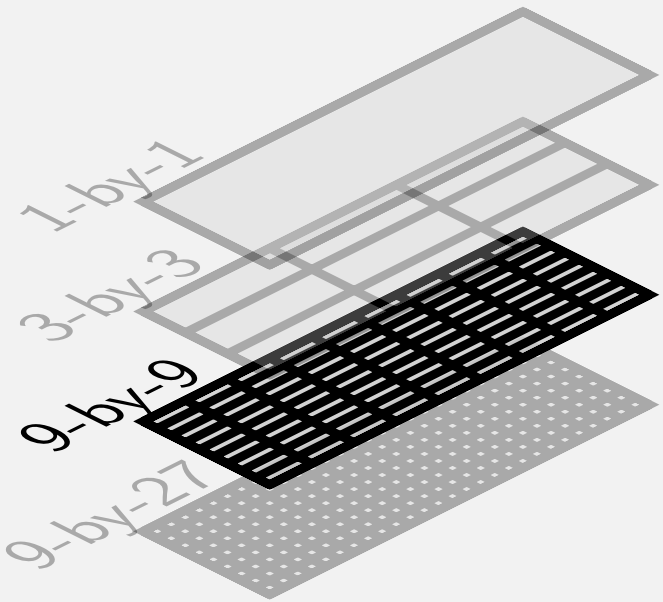
Step 2

$s = \text{seed}_1$

Block sparsity seeding with $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ results in $d = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

Obtaining sparsity (hierarchical)

Step 2

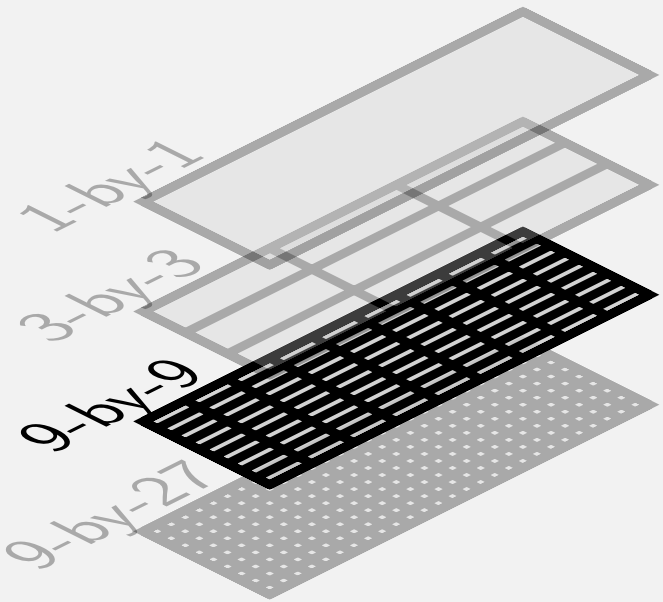


$$r = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \bigstar & \cdot \\ \cdot & \cdot & \cdot & \cdot & \bigstar & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \bigstar & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \bigstar & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \bigstar & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \bigstar & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \bigstar \\ \bigstar & \bigstar & \bigstar & \cdot & \cdot & \cdot & \bigstar & \bigstar & \bigstar \\ \cdot & \cdot & \bigstar & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

$$d = \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{array}\right]$$

Obtaining sparsity (hierarchical)

Step 2



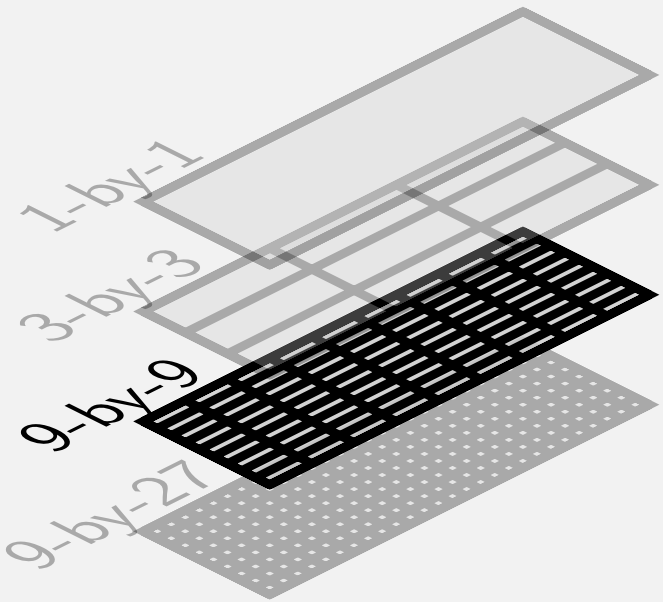
$$r = \begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \star & \cdot \\
\cdot & \cdot & \cdot & \cdot & \star & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \star & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \star & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\star & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \star & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \star \\
\star & \star & \star & \cdot & \cdot & \cdot & \star & \star & \star \\
\cdot & \cdot & \star & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}$$

$$\mathrm{col}(r) = \left\{ \begin{bmatrix}1\\0\\0\\1\\1\\1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix} \right\}$$

$$\mathrm{col}(r^T) = \left\{ \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\0\\1\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix} \right\}$$

Obtaining sparsity (hierarchical)

Step 2



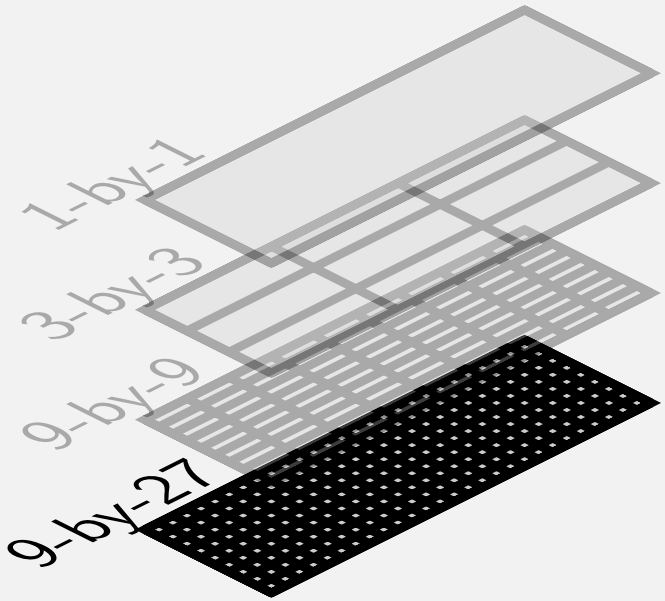
$$r = \left(\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right)$$

$$\text{seed} = \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right\}$$

$$\text{mode} = \text{adjoint}$$

Obtaining sparsity (hierarchical)

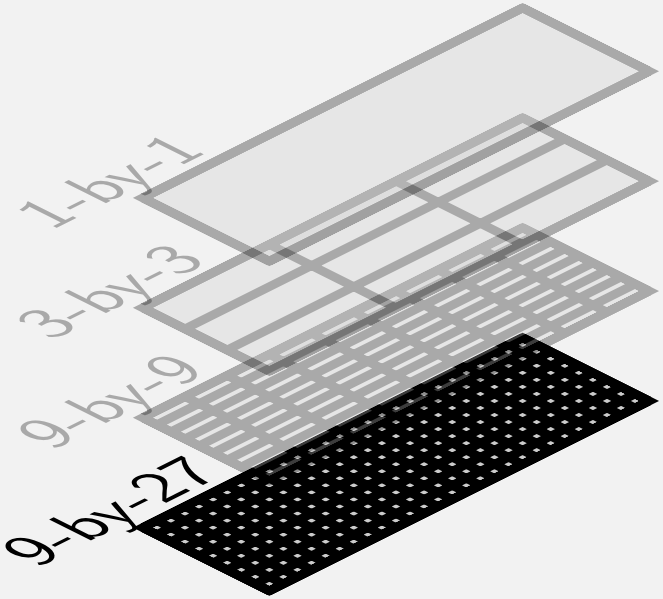
Step 3



$$r = \begin{pmatrix} ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \end{pmatrix}$$

Obtaining sparsity (hierarchical)

Step 3



$$r = \begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & ? & ? & ? & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
? & ? & ? & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? & \cdot \\
? & ? & ? & ? & ? & ? & ? & ? & ? & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? & ? & ? & ? & ? & ? \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ? & ? & ? & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}$$

Obtaining sparsity (hierarchical)

Step 3

For each s in seed

Reverse block sparsity seeding with $s \bigotimes \mathbf{1}^1 \bigotimes v^1$

Obtaining sparsity (hierarchical)

Step 3

$s = \text{seed}_0$

Reverse block sparsity seeding with $\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \otimes [1] \otimes [1] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$

Obtaining sparsity (hierarchical)

Step 3

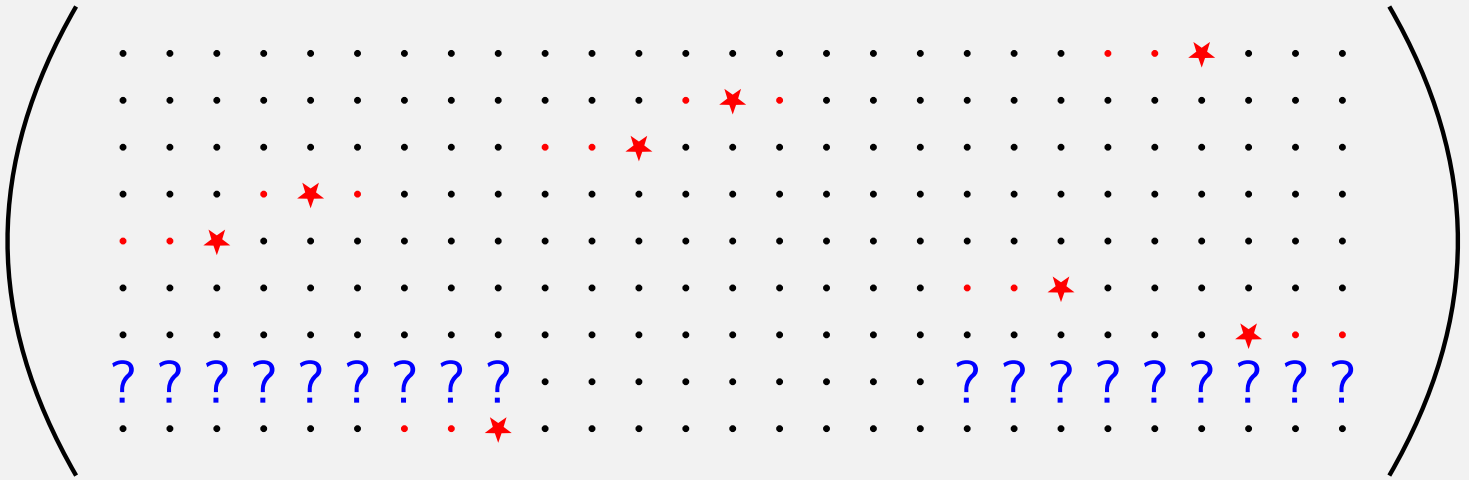
$s = \text{seed}_0$

Reverse block sparsity seeding with $\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\0\\1\end{bmatrix}$ results in $d = \begin{bmatrix}0\\0\\1\\0\\1\\0\\0\\0\\1\\0\\0\\1\\0\\1\\0\\0\\0\\0\\0\\1\\0\\0\\1\\1\\0\\0\end{bmatrix}$

Obtaining sparsity (hierarchical)

$$d = \left[0\ \ 0\ \ 1 \mid 0\ \ 1\ \ 0 \mid 0\ \ 0\ \ 1 \mid 0\ \ 0\ \ 1 \mid 0\ \ 1\ \ 0 \mid 0\ \ 0\ \ 0 \mid 0\ \ 0\ \ 1 \mid 0\ \ 0\ \ 1 \mid 1\ \ 0\ \ 0\right]^T$$

$r =$



Obtaining sparsity (hierarchical)

Step 3

$s = \text{seed}_0$

Reverse block sparsity seeding with $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \otimes [1] \otimes [1] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

Obtaining sparsity (hierarchical)

Step 3

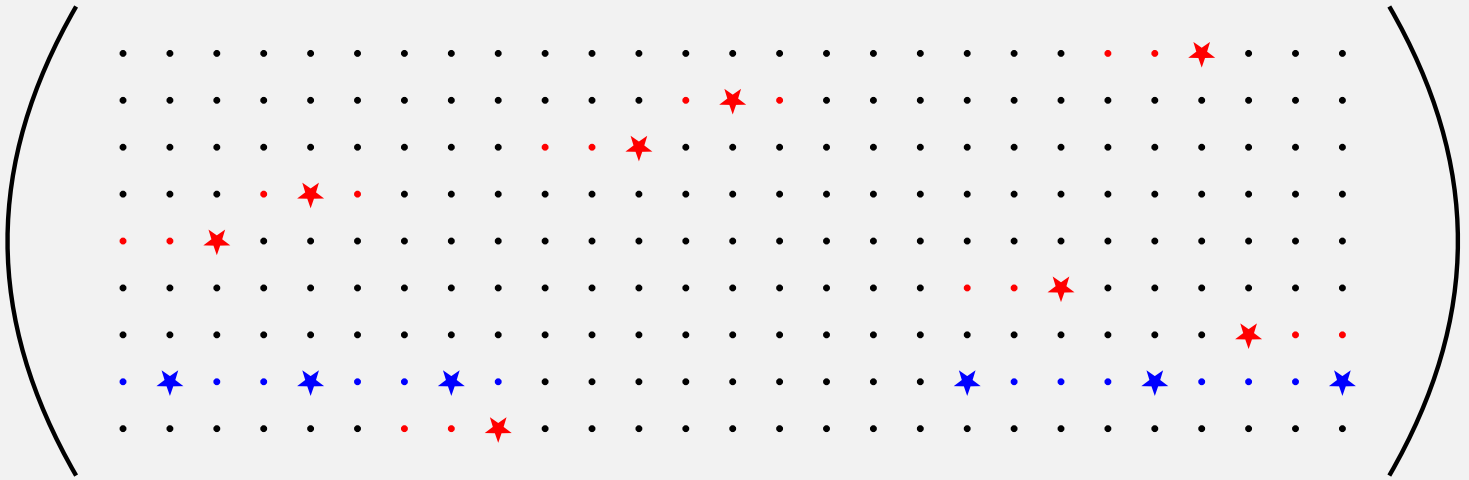
$s = \text{seed}_0$

Reverse block sparsity seeding with $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ results in $d = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

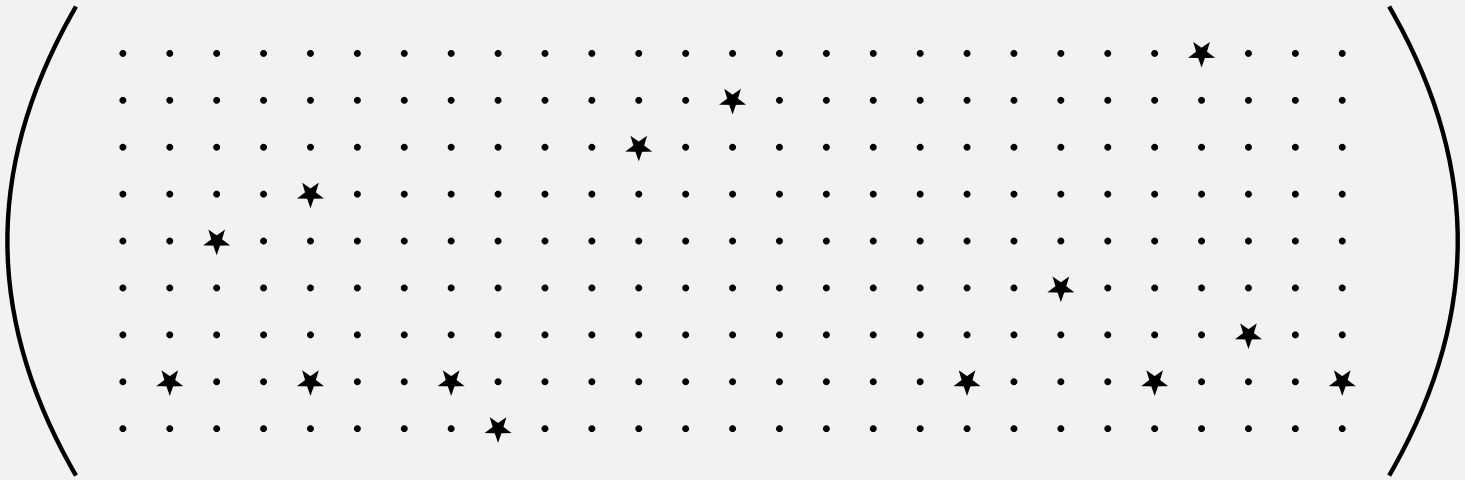
Obtaining sparsity (hierarchical)

$$d = \left[0 \;\; 1 \;\; 0 \,\middle|\, 0 \;\; 1 \;\; 0 \,\middle|\, 0 \;\; 1 \;\; 0 \,\middle|\, 0 \;\; 0 \;\; 0 \,\middle|\, 0 \;\; 0 \;\; 0 \,\middle|\, 0 \;\; 0 \;\; 0 \,\middle|\, 1 \;\; 0 \;\; 0 \,\middle|\, 0 \;\; 1 \;\; 0 \,\middle|\, 0 \;\; 0 \;\; 1\right]^T$$

$r =$



Obtaining sparsity (hierarchical)



Mission

Obtain the above sparsity with less than 9 dependency sweeps.

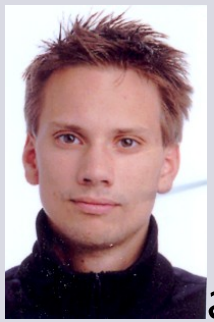
Debriefing

We actually used 11 dependency sweeps . . .

Table of Contents

Implementation

www.casadi.org CasADi

- Low-level tool for quick, yet highly efficient implementation of algorithms for *dynamic optimization*
- LGPL-licensed
- C++ / C++11
- Interfaces to Python, Haskell
- Numerical backends: IPOPT, SNOPT, WORHP, Sundials
- Coded by  and  , (2010-present)

Implementation

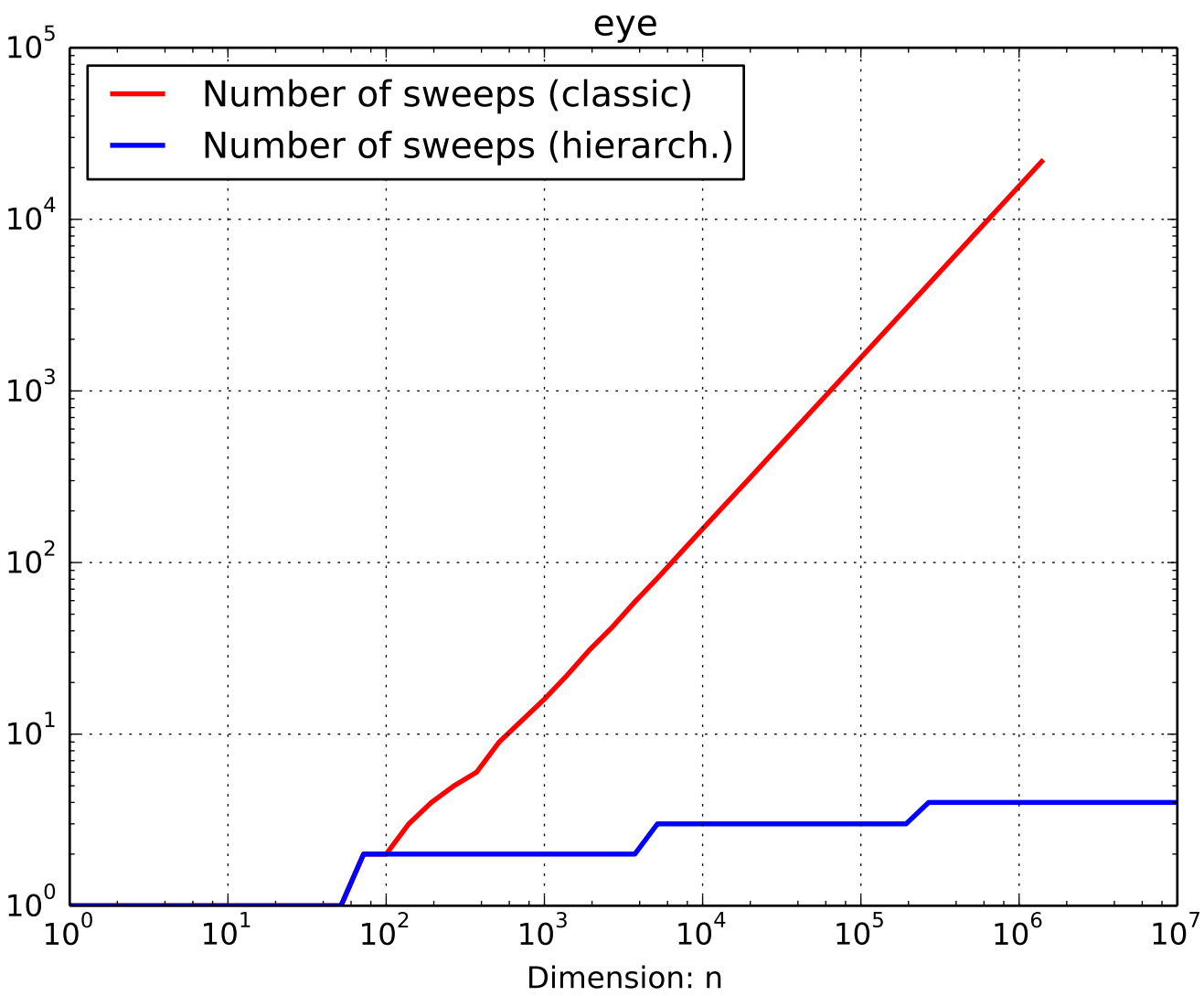
- Implemented for arbitrary matrix sizes
- CasADi does 64 dependency sweeps in parallel; choice: $\sigma = 64$
- Adaption for star-coloring

Numerical experiments

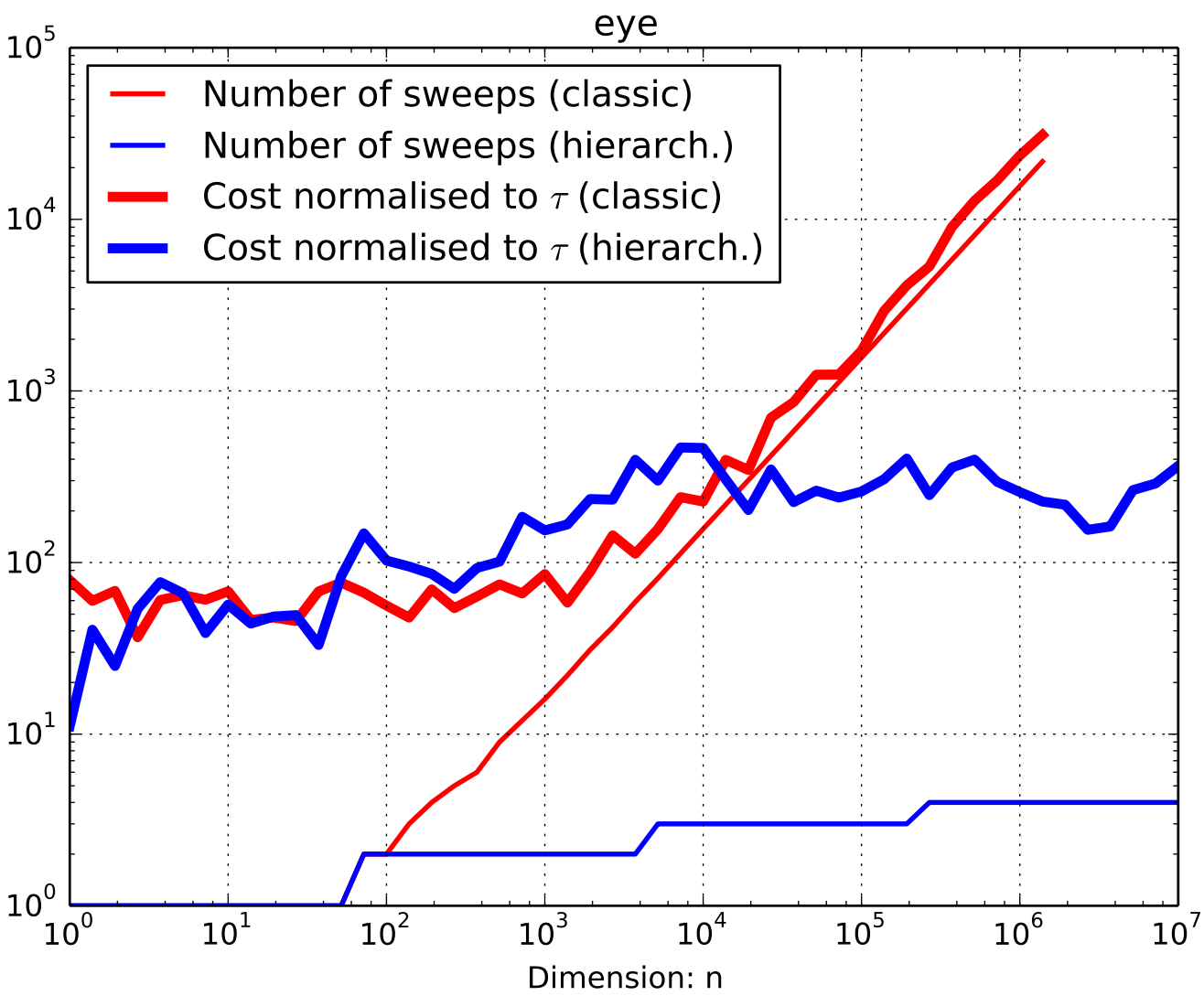
Setup

- $f(x) = Ax, \qquad A \in \mathbb{R}^{n \times n}$
- Try different sparsities for A

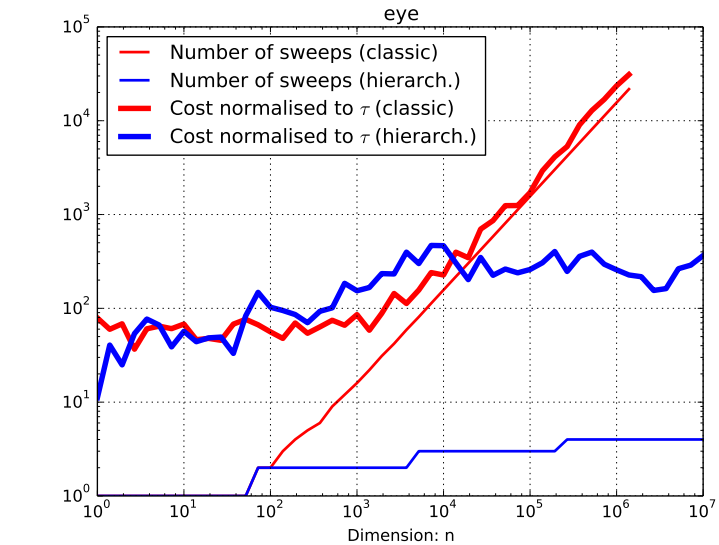
Numerical experiments



Numerical experiments



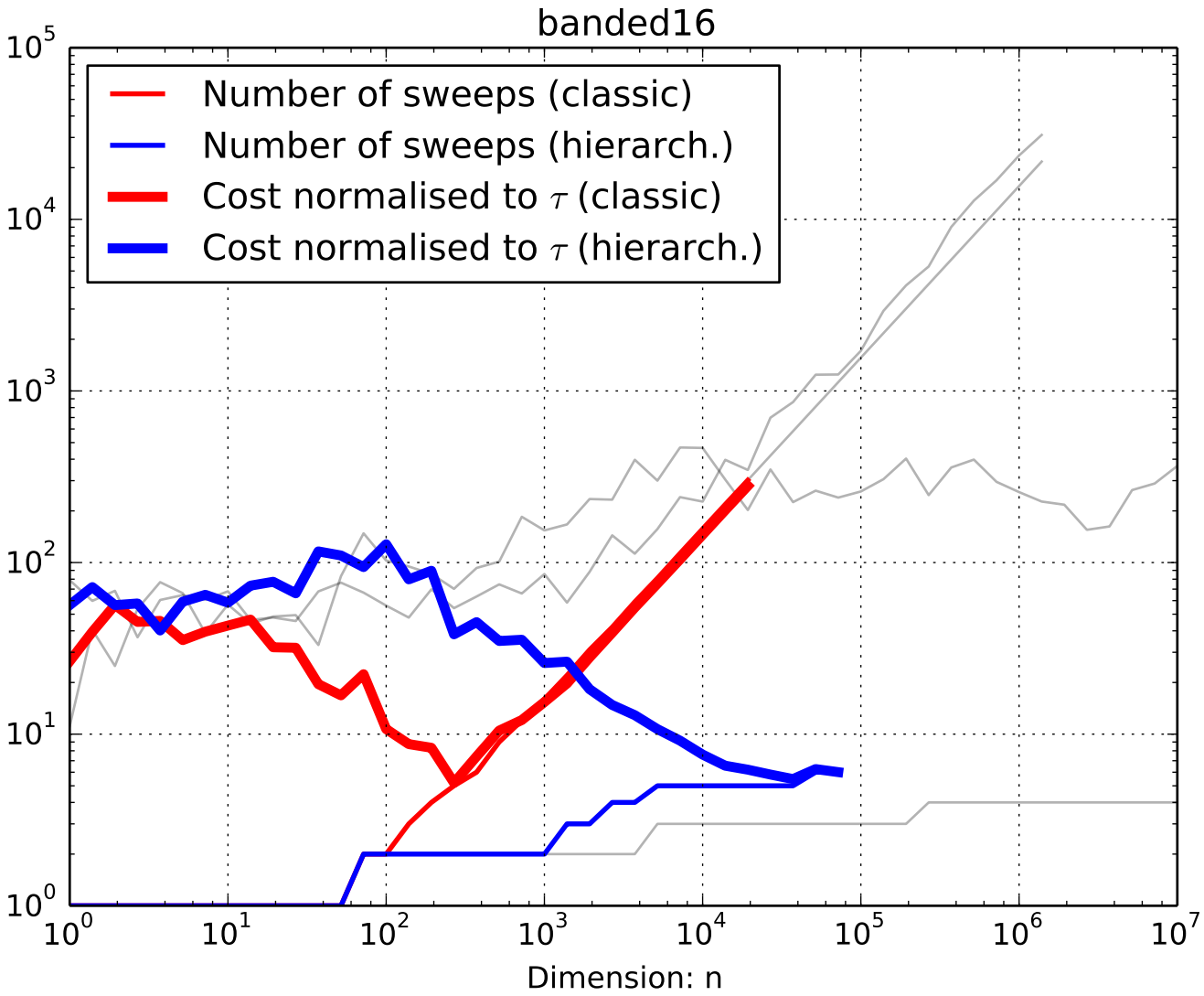
Numerical experiments



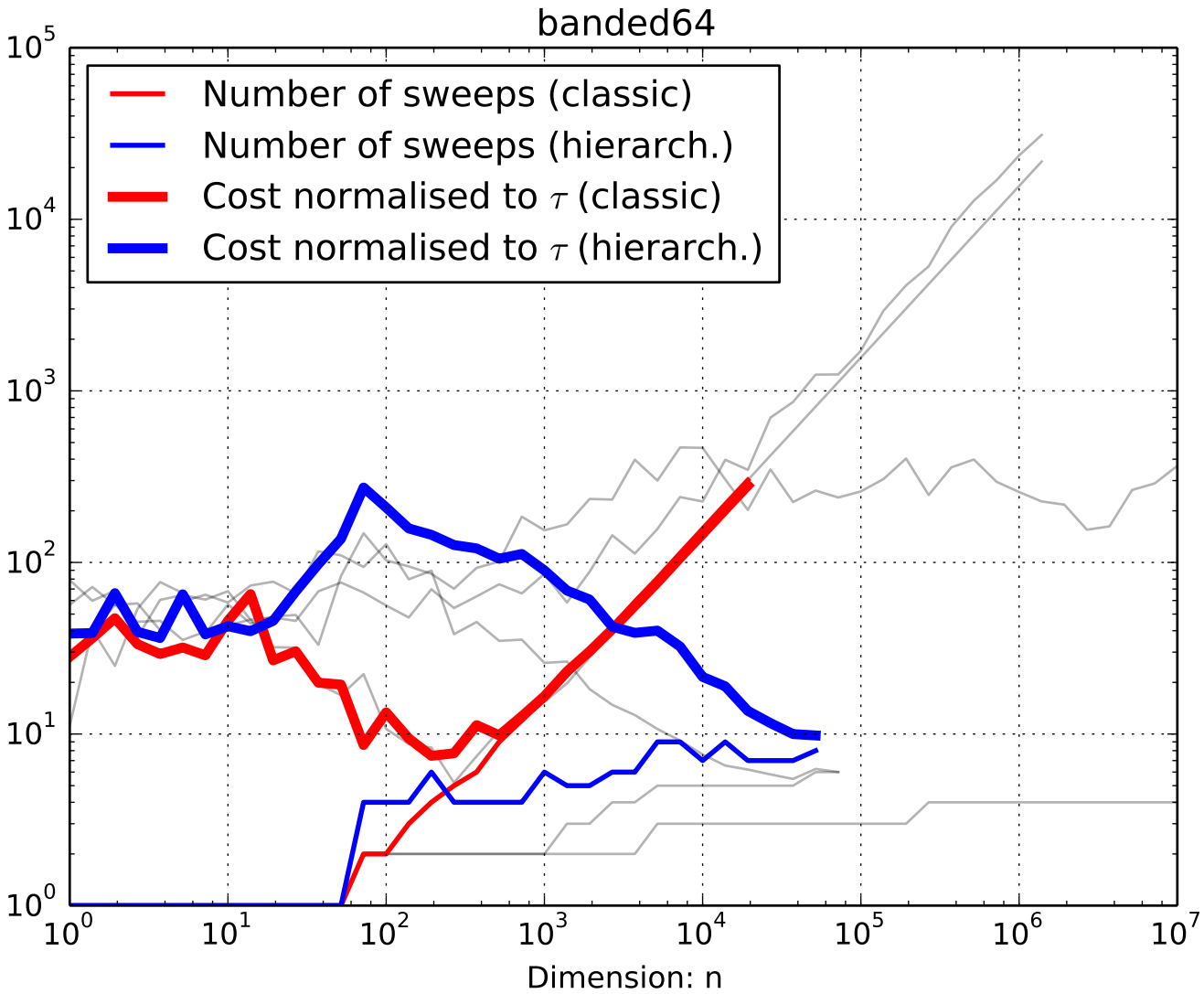
Mind the gap

- Cost of the algorithm itself
- Most notable: coloring

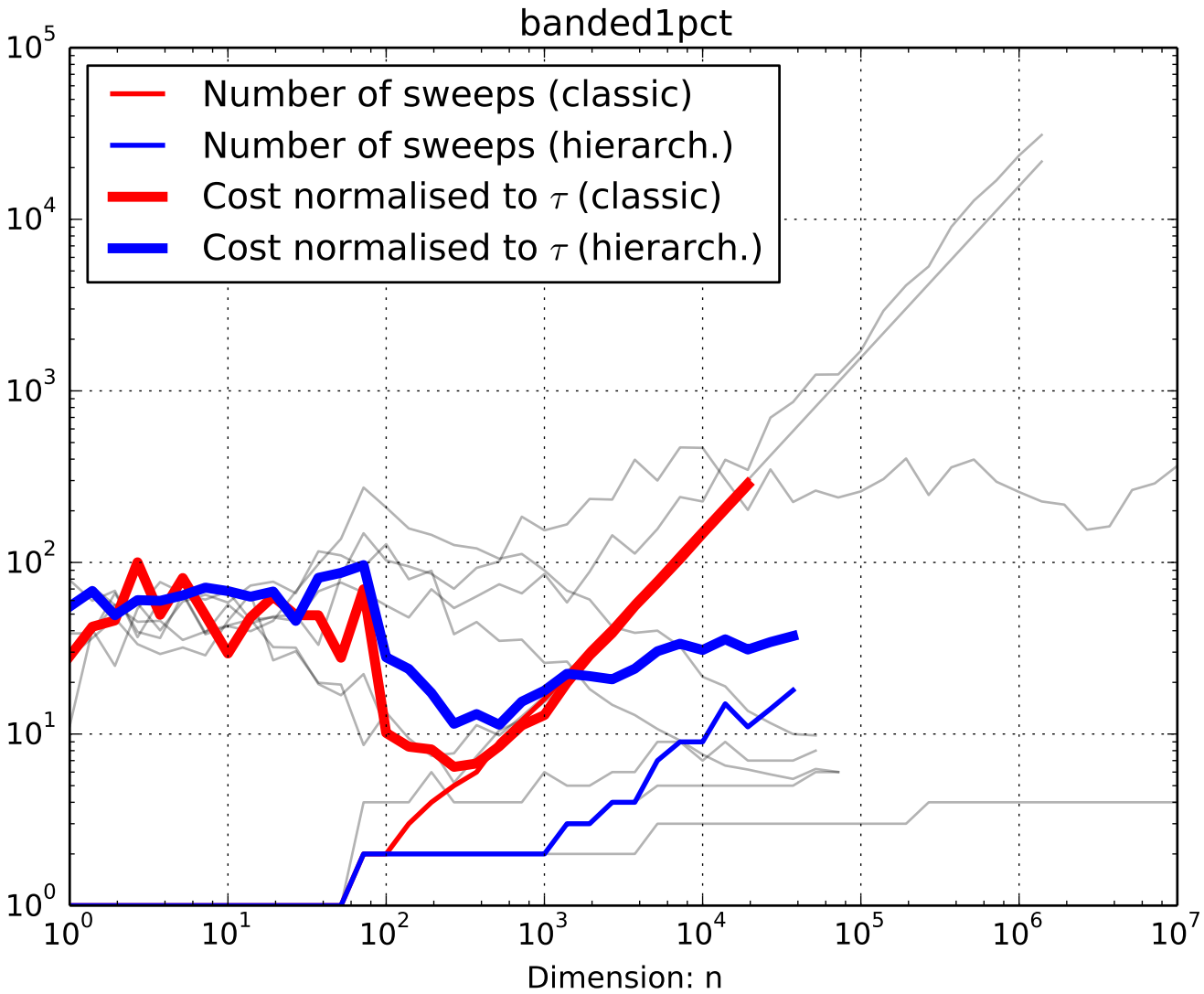
Numerical experiments



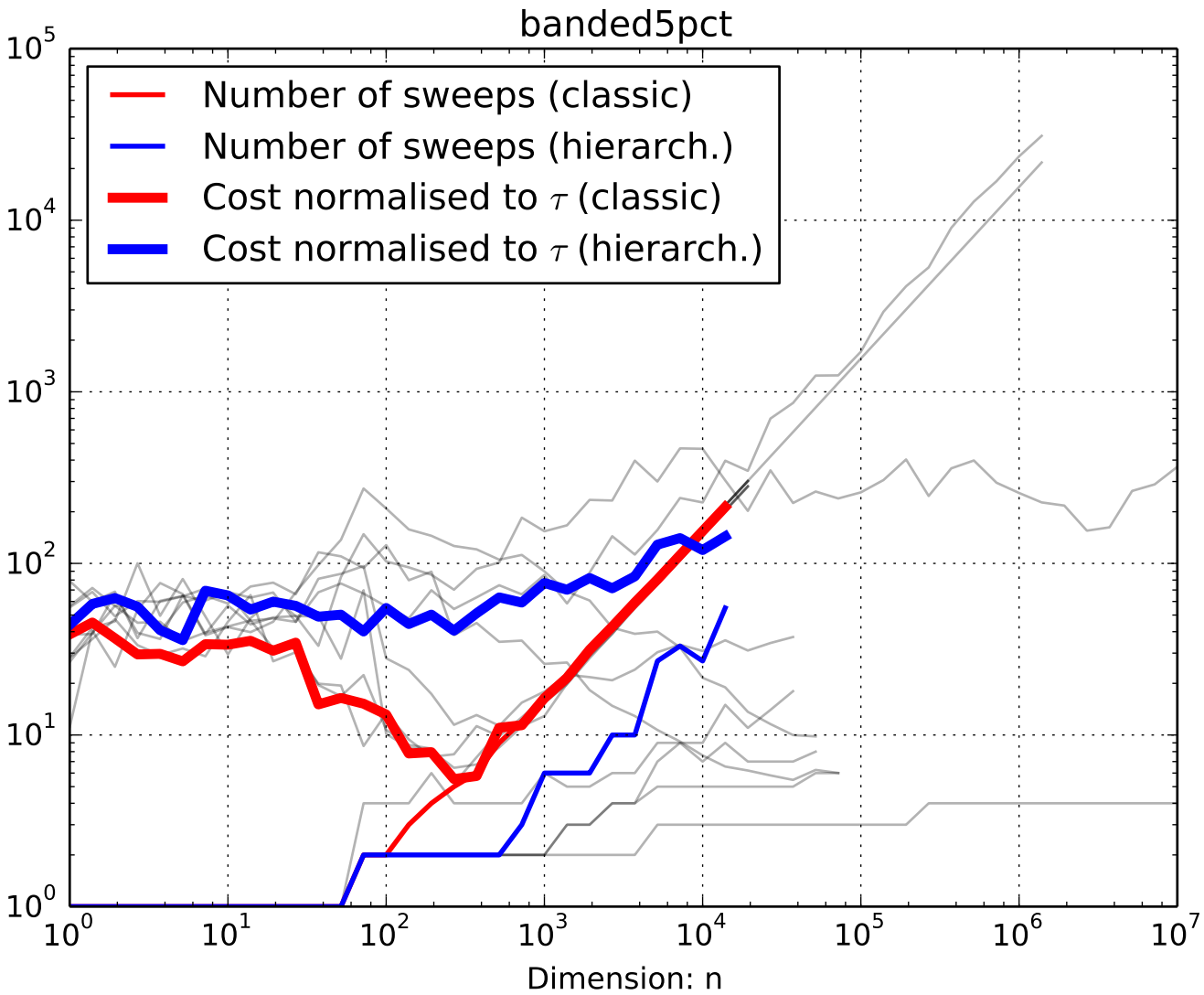
Numerical experiments



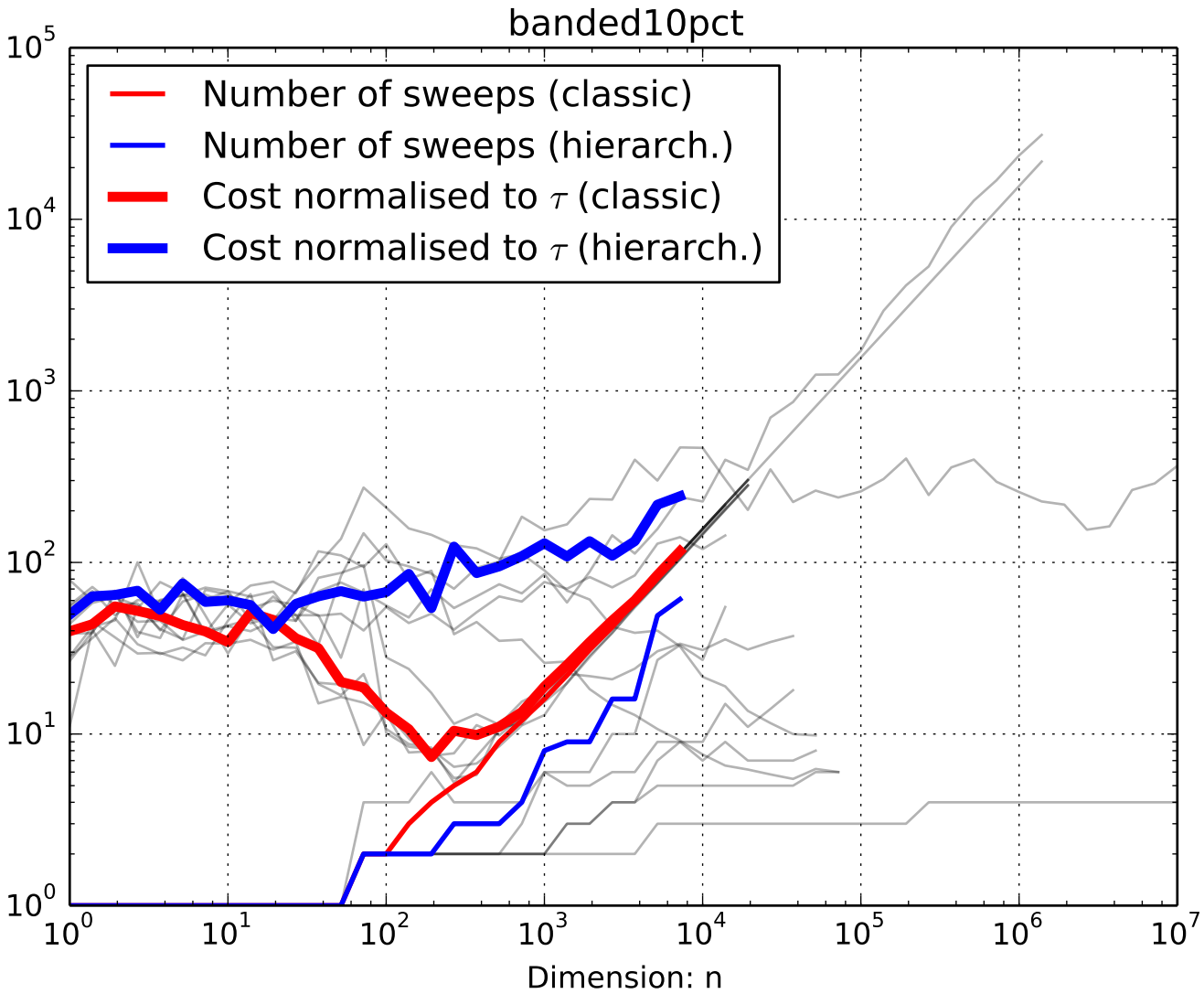
Numerical experiments



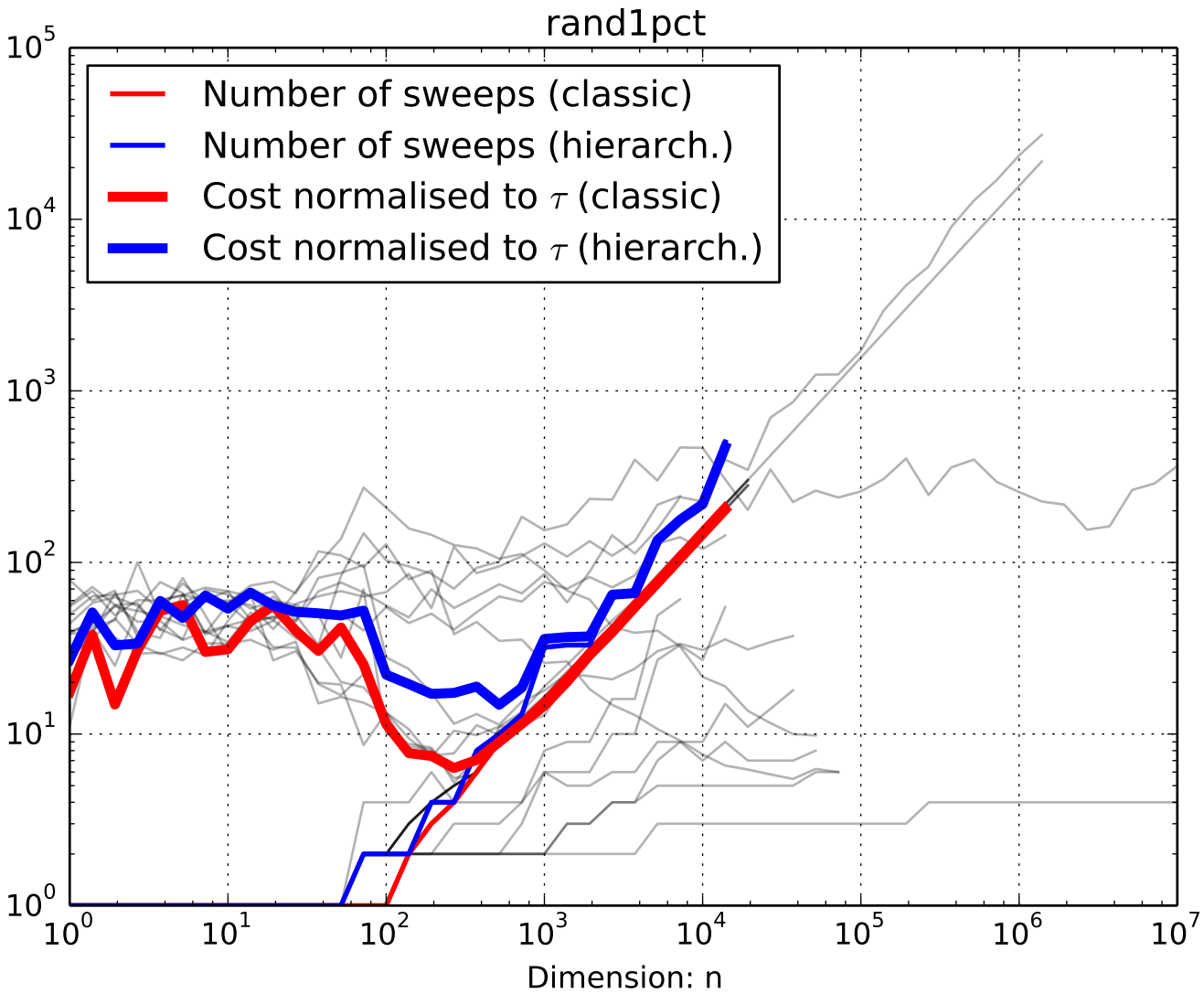
Numerical experiments



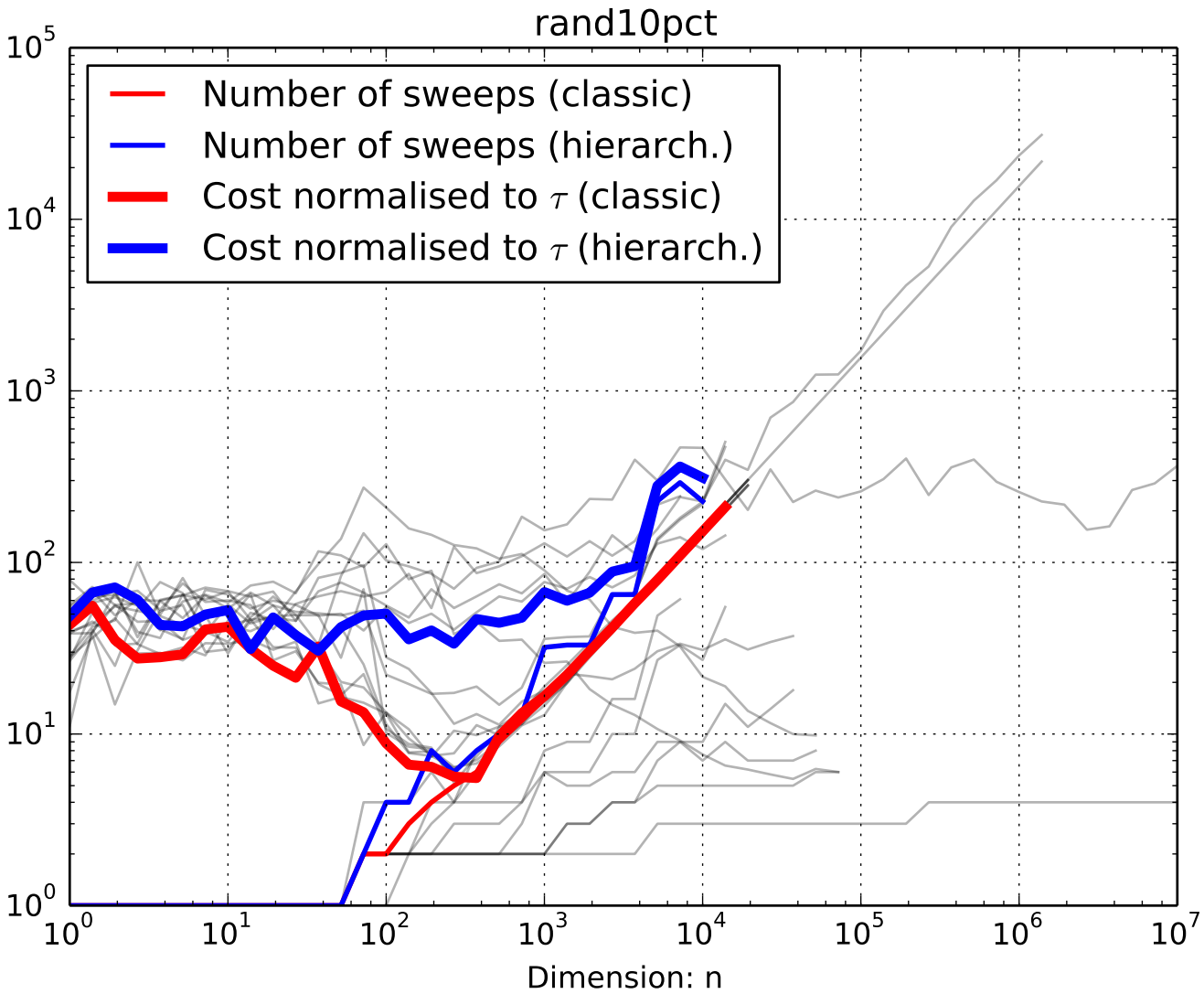
Numerical experiments



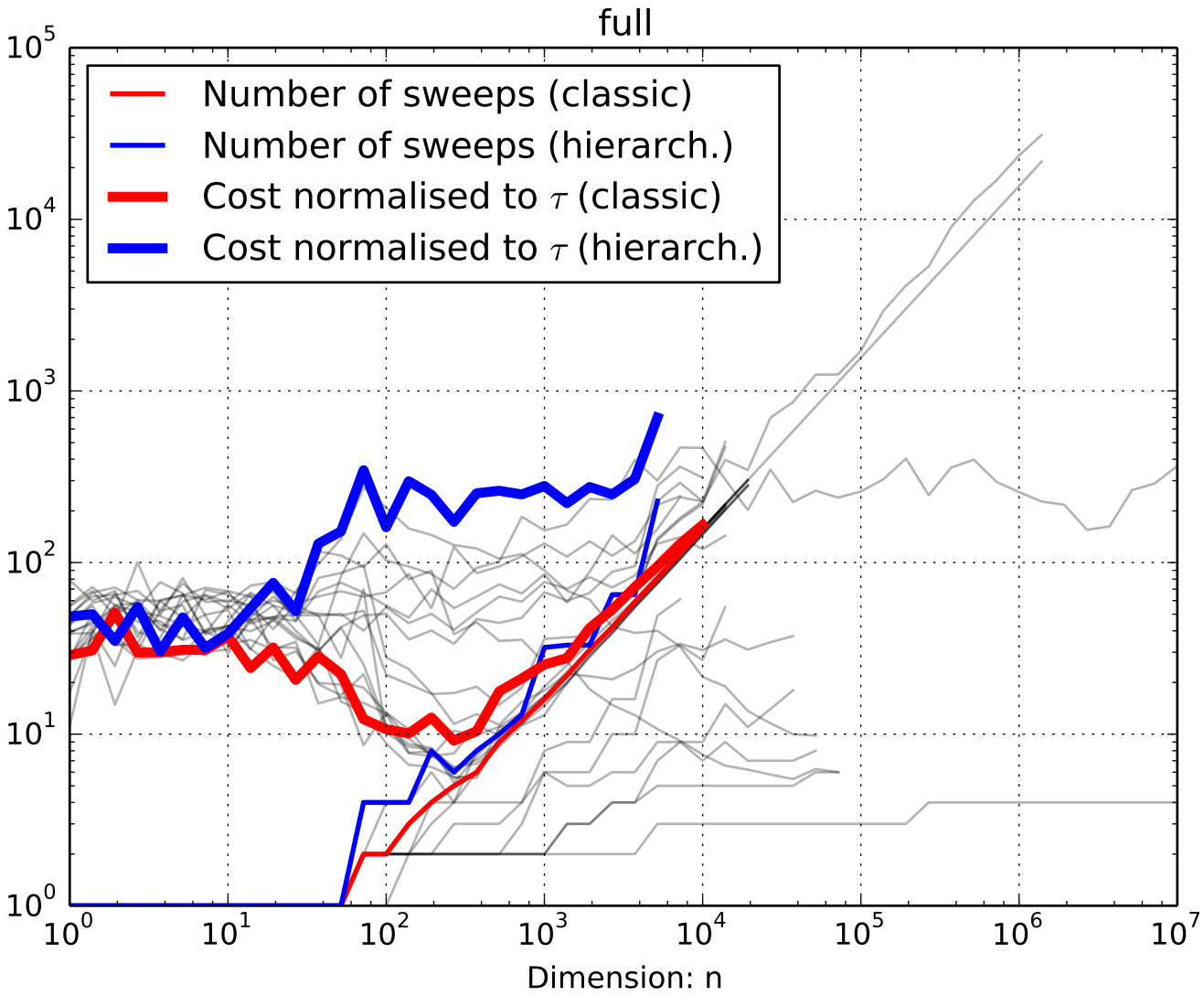
Numerical experiments



Numerical experiments



Numerical experiments



Conclusion

When does hierarchical sparsity detection shine?

- n *and* m large
- τ substantial: complex $f(x)$
- Highly structured

Typical case

- Optimal control problems solved with direct collocation